VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020 Bc. Josef Cigánek

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

ZÁKLADNÍ VLASTNOSTI SÍŤOVÝCH PROTOKOLŮ A KOMUNIKAČNÍCH TECHNIK
BASIC PROPERTIES OF NETWORK PROTOCOLS AND COMMUNICATION TECHNIQUES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Josef Cigánek**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Jan Dvořák**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Josef Cigánek ***ID:*** 174174
Ročník: 2 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Základní vlastnosti síťových protokolů a komunikačních technik

POKYNY PRO VYPRACOVÁNÍ:

V rámci Diplomové práce nastudujte fungování základních síťových protokolů a technologií. Především se zaměřte na protokoly typu TCP (Transmission Control Protocol), UDP (User Datagram Protocol), ARP (Address Resolution Protocol), IP (Internet Protocol), nebo komunikační techniky typu unicast, multicast, případně broadcast. Také se zaměřte na programy typu ping a traceroute. Navrhněte a podrobně popište dvě laboratorní úlohy zahrnující některé z uvedených protokolů a technologií ve vhodném simulačním či virtuálním prostředí. Po konzultaci s vedoucím mohou být použity i jiné komunikační protokoly. Výstupem této práce budou dvě laboratorní úlohy, zabývající se základními vlastnostmi síťových protokolů a technologií, spolu s podrobnými návody pro studenty komunikačních technologií. Vypracované návody budou prokonzultovány s vedoucím práce a budou obsahovat případné předpřipravené situace, doplňující úkoly, případné otázky na studenty a vypracované vzorové řešení úloh. Návody budou vypracovány v češtině a maximální délka pro vypracování jedné úlohy bude dvě hodiny času.

DOPORUČENÁ LITERATURA:

[1] JEŘÁBEK, J. Komunikační technologie. Skriptum FEKT Vysoké učení technické v Brně, 2019. s. 1-175.

[2] FOROUZAN, B. A. TCP/IP Protocol Suite. Fourt hedition, Boston: McGraw-Hill Higher Education, 2010, 979 stran. ISBN 978-0-07-337604-2.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 1.6.2020

Vedoucí práce: Ing. Jan Dvořák

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Diplomová práce je zaměřena na základní vlastnosti síťový protokolů. Hlavním úkolem je vytvořit laboratorní úlohy, které se zabývají základními vlastnostmi síťových protokolů a komunikačních technik, spolu s podrobnými návody pro studenty komunikačních technologií. Teoretická část práce seznamuje čtenáře s tématem počítačových sítí a se základními síťovými protokoly architektury TCP/IP, které se vyskytují v rámci vytvořených laboratorních úloh. Praktická část obsahuje výběr testovacího prostředí a následný popis instalace a přípravy vybraného prostředí GNS3. Závěrem jsou představeny vytvořené laboratorní úlohy. První úloha se věnuje základním rozdílům mezi transportními protokoly TCP a UDP. Druhá úloha je zaměřená na problematiku skupinového vysílání a třetí se věnuje základům penetračního testování.

KLÍČOVÁ SLOVA

GNS3, TCP/IP, síťové protokoly, laboratorní úlohy

ABSTRACT

The diploma thesis is focused on the basic properties of network protocols. The main task is to create laboratory tasks that deal with the basic features of network protocols and communication techniques, along with detailed instructions for students of communication technologies. The theoretical part of thesis acquaints the reader with the topic of computer networks and the basic network protocols of the TCP/IP architecture, which occur within the created lab. The practical part contains a selection of testing environment and subsequent description of installation and preparation of selected GNS3 environment. Finally are presented labs, which have been created. First lab deals with the basic differences between TCP and UDP. The second lab is focused on the issue of multicast and the third deals with the basics of penetration testing.

KEYWORDS

GNS3, TCP/IP, network protocols, labs

CIGÁNEK, Josef. *Základní vlastnosti síťových protokolů a komunikačních technik*. Brno, 2020, 128 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Jan Dvořák

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Základní vlastnosti síťových protokolů a komunikačních technik“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Janu Dvořákovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále své rodině a přítelkyni za podporu při celém studiu.

Brno ....

podpis autora

Obsah

Úvod

Historie počítačových sítí je úzce spjata s vývojem výpočetní techniky. Postupným vývojem počítačů, se naskytla otázka, jakým způsobem zajistit jejich vzájemnou komunikaci. Vznik první počítačové sítě se datuje rokem 1969, kdy se vědcům z Kalifornské univerzity podařila sestavit první experimentální počítačová síť. První počítačová síť dostala název Arpanet.

V osmdesátých letech minulého století se již začaly objevovat rozsáhlejší počítačové sítě a proto vznikla potřeba počítačové sítě určitým způsobem sjednotit a standardizovat. Jako první se o to pokusila organizace ISO, která v roce 1984 veřejně standardizovala sedmivrstvý síťový model, zvaný referenční model ISO/OSI. Nevyhovující vlastnosti a předpoklady navrženého modelu se postupem času ukázaly jako nevýhodné pro praktické použití a to hned z několika důvodů. Mezi hlavní důvody patřilo především upřednostňování spojovaných přenosů, opakované funkce na jednotlivých vrstvách a důraz kladený především na síťové prvky, nikoliv na koncové stanice.

Síťová architektura, které se podařilo prosadit na celosvětové úrovni, nese označení architektura TCP/IP. Architektura TCP/IP vznikla za spolupráce agentury ARPA a ministerstva obrany USA. Jedná se o soustavu síťových protokolů, které pracují na 4 vrstvovém systému. Architektura TCP/IP je používaná dodnes a na jejím principu je postavena celosvětová síť Internet.

První část diplomové práce se zabývá základními typy a principy komunikace v počítačových sítích, referenčním modelem ISO/OSI a architekturou TCP/IP. Následně budou představeny základních protokoly TCP/IP, které jsou využity v rámci laboratorních úloh. Protokoly jsou rozdělené dle vrstev, na kterých pracují. Při popisu ICMP protokolu jsou popsány i programy typu PING a Traceroute, které se velmi často používají v rámci diagnostiky počítačových sítí.

V praktické části jsou uvedena vhodná testovací prostředí pro tvorbu laboratorních úlohy. Dále je zde uveden postup instalace zvoleného simulačního prostředí, kterým je prostředí GNS3. Na závěr jsou představeny 3 vytvořené laboratorní úlohy. První úloha se věnuje základním rozdílům mezi transportními protokoly TCP a UDP. Druhá úloha je zaměřená na problematiku skupinového vysílání a třetí se věnuje základům penetračního testování. V příloze práce se nachází vypracovaný návody pro studenty k vytvořeným laboratorním úlohám.

1 Komunikace v počítačových sítích

Pod pojmem počítačová síť [1] si je možné představit programové a technické prostředky, které zprostředkovávají spojení a vzájemnou komunikaci mezi síťovými entitami. Aby počítačová sít mohla vzniknout, musí být mezi těmito entitami navázána datová komunikace pomocí vybraného telekomunikačního systému. Pod slovním spojením počítačová síť by se mohlo mylně zdát, že se jedná pouze o počítače, které jsou mezi sebou propojené. Síťové entity, které byly v předešlých definicích počítačové sítě zmíněny, však reprezentuje celá řada různých síťových prvků, které jsou schopny určitým způsobem pracovat s procházející datovou komunikací. V kapitole 1.2 budou základní síťové prvky představeny [2, 3].

1.1 Rozdělení počítačových sítí

Počítačové sítě je možné dělit dle mnoha kriterií. Jedním z nejčastějších dělení je rozlišení dle velikosti dané sítě, použité topologie a také dle dané hierarchie prvků v síti.

1.1.1 Rozdělení dle velikosti sítě

PAN (Personal Area Network)

Jedná se o nejmenší možnou definovanou počítačovou síť, jež je využívána především pro osobní použití. Využívá se přenos dat většinou mezi mobilními zařízeními a to na velmi krátké vzdálenosti, maximálně do 10 metrů [4]. Ve většině případů se jedná o přenos souborů pomocí bezdrátové technologie bluetooth nebo kabelovým spojením přes USB porty.

LAN (Local Area Network)

Patří do kategorie menších sítí, které svojí rozlohou zabírají geografickou oblast v řádech jednotek kilometrů. Typicky se jedná o domácí či firemní síť připojující uživatele v rámci jedné budovy. V sítích LAN se nejčastěji používá technologie Ethernet nebo Wi-fi [5].

MAN (Metropolitan Area Network)

Metropolitní typ sítě pokrývající oblasti na úrovni měst. Sítě MAN obvykle poskytují internetové připojení menším sítím LAN. Rychlost přenosu se velmi často pohybuje v jednotkách Gb/s [6].

WAN (Wide Area Network)

Polohou nejrozsáhlejší typy počítačových sítí, které svou rozlohou pokrývají oblasti států či kontinentů. Protože se jedná o komunikaci na velmi dlouhé vzdálenosti s velkým množstvím připojených subjektů, dochází zde k vyššímu zpoždění přenosu. V sítí WAN se můžeme setkat například s technologiemi ATM (Asynchronous Transfer Mode), MPLS (Multiprotocol Label Switching) nebo FR (Frame Relay). Nejznámější a nejpoužívanější propojení WAN sítí známe jako síť Internet [5].

1.1.2 Rozdělení sítí dle topologie

Dalším kriteriem, dle kterého lze počítačové sítě rozlišovat, je typ síťové topologie. Mezi nejznámější topologie počítačových sítích patří [7, 8]:

Hvězda

Všechny uzly sítě jsou propojeny s centrálním uzlem, který veškerou komunikaci řídí. Na tomto centrálním rozbočovači závisí veškerý výkon sítě a pokud dojde k jeho výpadku, ztratí spojení všechny připojené uzly. Na druhou stranu je tato topologie snadno rozšiřitelná o nové prvky, poskytuje snazší správu a vysoké rychlosti [7].

Kruh

Jak napovídá název, výsledné zapojení tvoří tvar kruhu a proto je každý uzel propojený pouze se sousedním uzlem. Poměrně jednoduchá technologie, kde je tok dat posílán pouze v jednom směru a tak se snižuje pravděpodobnost případné kolize. K řízení toku dat není zapotřebí žádný řídící prvek. Avšak jakékoliv přerušení spojení, či vypadnutí jediného prvku, má za následek narušení chodu celé sítě [7].

Strom

Podobně jako u topologie hvězda, jsou všechny uzly postupně připojeny k jednomu centrálnímu uzlu. U stromové topologie jsou však prvky propojeny hierarchicky, jakožto větve stromu. To vede k škálovatelnosti prvků a vetší flexibilitě. Stromové topologie poskytují snazší údržbu a identifikaci poruch za cenu obtížnější konfigurace. Pokud dojde k výpadku uzlu, rozpadne se podstrom, který je navázaný za tímto uzlem [7].

Úplný polygon

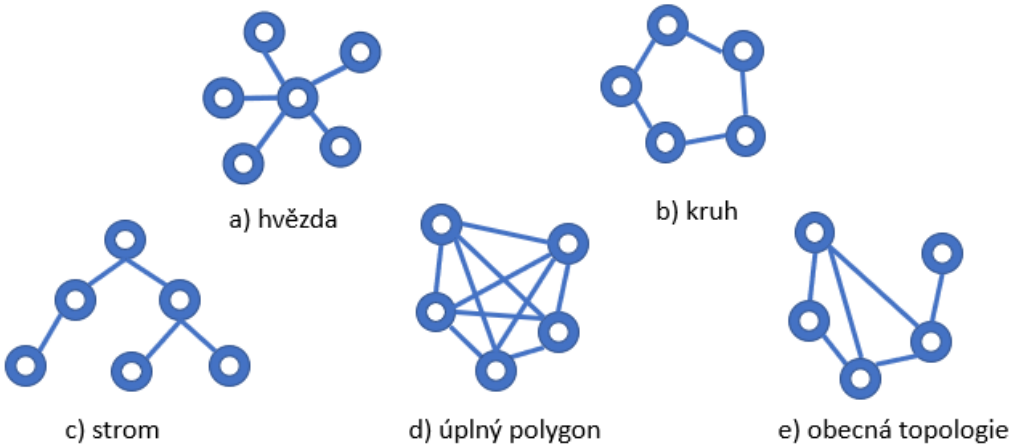
Všechny uzly sítě jsou mezi sebou navzájem propojeny přímým spojem. Maximální efektivita této topologie vyžaduje vysoké náklady na realizaci, které rostou s každým

přidaným prvkem topologie [8].

Obecná topologie

Jedná se o topologii připomínající neúplný polygon. Obecná topologie má nepravidelný počet síťových propojů mezi uzly. Na uzlech, kde se očekává vyšší datový tok, se nachází větší počet propojů kvůli zachování redundantnosti [8].

V reálných sítích se síťové topologie často kombinují v závislosti na požadované vlastnosti dané sítě. Grafické znázornění výše uvedených topologii se nachází na obrázku 1.1.



Obr. 1.1: Základní typy síťových topologií [5].

1.1.3 Rozdělení sítí dle hierarchie prvků

Počítačové sítě se dají rozdělit také dle komunikační úrovně a hierarchie daných prvků:

- **Klient-server** - Architektura s rozdílnou úrovní hierarchie prvků v konkrétní počítačové síti. Architektura je založena na principu posílání žádostí klientů o služby, jež poskytuje strana serverů. Entity vyšší úrovně (servery) zpracovávají žádosti a v závislosti na nastavených politikách a pravidlech rozhodují o tom, jestli klientovi danou službu poskytnou, nebo ne [9].
- **Klient-klient** (peer to peer) - Architektura, kde si jsou všechny prvky sítě rovny. Odpadá tak veškerá hierarchie mezi prvky. Všechny prvky počítačové sítě spolu tedy mohou komunikovat na přímo [9].

1.2 Základní síťové prvky

Síťové prvky můžeme rozdělit do dvou skupin. První skupinou jsou tzv. pasivní síťové prvky. Prakticky se jedná o síťové spoje, mezi které řadíme především síťovou kabeláž, konektory, spojky, rozvaděče nebo zásuvky. Pasivní síťové prvky pouze data přenáší a žádným způsobem do komunikace nezasahují [8]. Druhou skupinou jsou aktivní prvky. Ty již do procházející datové komunikace aktivně zasahují a aktivně ji ovlivňují [5]. Níže následuje výpis v současnosti nejpoužívanějších aktivních síťových prvků:

Opakovač (Repeater)

Zařízení, jehož jedinou funkcí je přijatý signál, který dorazí na vstupní port obnovit či zesílit a výstupním portem odeslat dále do sítě. V dnešní době se opakovače používají převážně u optických a bezdrátových sítí [8, 9].

Rozbočovač (HUB)

Stejně jako opakovač, tak i rozbočovač pracuje maximálně na fyzické vrstvě. Na rozdíl od předešlého prvku, je rozbočovač typicky osazen více než dvěma porty. Veškerá data, která dorazí na jakýkoliv port rozbočovače jsou přeposlána na všechny zbývající porty. Data se tak poměrně rychle dostávají ke všem připojeným prvkům, kterým ale původně nemusela být adresována a dochází tak k zbytečnému zahlcování sítě. Dnes se s rozbočovači můžeme setkat například v sítích PAN, kde se velmi často používají USB rozbočovače [8, 9].

Přepínač (Switch)

Na rozdíl od rozbočovače jde o poznání inteligentnější prvek. Přepínač již dokáže rozlišit přesného adresáta a přicházející data tak posílá jen na konkrétní port, ke kterému je skutečný příjemce připojen. Správného příjemce přepínač zjišťuje dle MAC (Media Access Control) adresy[9]. Přepínač si vede tabulku MAC adres všech zařízení, které jsou k jeho portům připojeny a na základě těchto informací volí správné porty. Tímto tedy nedochází k zbytečnému zahlcování sítě, jako v případe rozbočovače.

Základní typy přepínačů pracují na linkové vrstvě ISO/OSI modelu. Existují ovšem i přepínače, které umí pracovat i na vyšších vrstvách. L3 přepínače dokáží navíc i analýzu záhlaví protokolů síťové vrstvy, což může být výhodné pro zvýšení výkonu a efektivity přepínaní paketů, ale také pro určité zajištění zabezpečení a kvality služeb (QoS)[8]. Ve firemních prostředí bývají L3 přepínače také velmi

často využívány pro vytváření podsítí VLAN (Virtual LAN), což umožňuje administrátorům logické oddělení jednotlivých virtuálních sítí v rámci jedné sítě LAN. Mezi přepínače vyšších vrstev patří také přepínače L4 a MLS. Tyto přepínače, které dosahují až do aplikačních vrstev, naleznou využití například při vyrovnávání zátěže mezi aplikačními servery nebo pro vykonávání překladu IP adres a aplikačních portů [8, 9, 12].

Směrovač (Router)

Aktivní síťový prvek pracující na síťové vrstvě ISO/OSI, jehož hlavní funkcí je směrování. Směrování je proces, při kterém jsou pakety na základě cílové IP adresy a informací z hlavičky paketu, odesílány na cílové směrovače v jiných sítích [8]. Pro zajištění korektního směrování si routery vedou tzv. směrovací tabulku. Směrovací tabulka obsahuje následující informace [12]:

- IP adresa sítě adresáta,
- Síťová maska,
- Výchozí brána,
- Rozhraní,
- Metrika,
- Použitý směrovací protokol.

Sousední routery si mezi sebou vyměňují informace, které vedou k ujasnění si obrazu topologie sítě a efektivnímu směrování. Využíváno je u toho i různých směrovacích protokolů (RIP a OSPF) [8].

Firewall

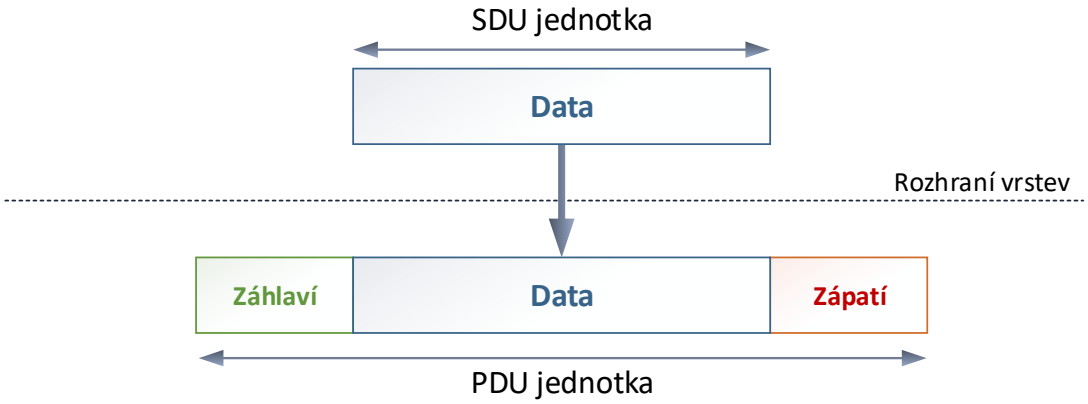
Bezpečnostní síťový prvek, který slouží k řízení a filtraci síťového provozu. Filtrace funguje pomocí nadefinovaných pravidel, kde je jasně stanoveno, které pakety mohou do sítě vstoupit a která ne [10]. Firewall tedy od sebe odděluje sítě s různou úrovní důvěryhodnosti. Firewally se dělí na [8, 10]:

- Nestavový firewall (paketový filtr),
- Stavový firewall (stavový paketový filtr),
- Aplikační brány,
- Firewally s kontrolou protokolů a IDS.

Nejnovější verze firewallů mají v sobě zakomponovány mnoho nových funkcí zvyšující celkové zabezpečení sítě. Příkladem jsou systémy detekce očekávaných útoků, databáze známých signatur útoku nebo funkce antiviru [8, 10].

1.3 Referenční model ISO/OSI

Referenční model ISO/OSI [11] byl v minulost první snahou o určitou standardizaci a určité teoretické sjednocení poznatků a navržených principů o tom, jak by se měli síťové komunikace tvořit k zajištění vzájemného fungování. Model je založený na vrstvovém systému, kde každá vrstva vykonává nadefinovanou funkci. Na popis funkcí jednotlivých vrstev je ISO/OSI model zaměřený nejvíce [5]. Jedná se o vertikální vrstvení, což znamená, že jednotlivé vrstvy leží v těsné blízkosti svých sousedních vrstev, se kterými dokáží komunikovat a tím si mezi sebou poskytovat danou službu. Základní jednotkou každé jednotlivé vrstvy ISO/OSI je tzv. jednotka PDU (Protocol Data Unit). Jednotka PDU má v sobě již zakomponované záhlaví dané vrstvy (případně i zápatí) obsahující potřebné režijní metadata (viz obrázek 1.2). Referenční model ISO/OSI tak funguje na postupném zapouzdření, či odpouzdření



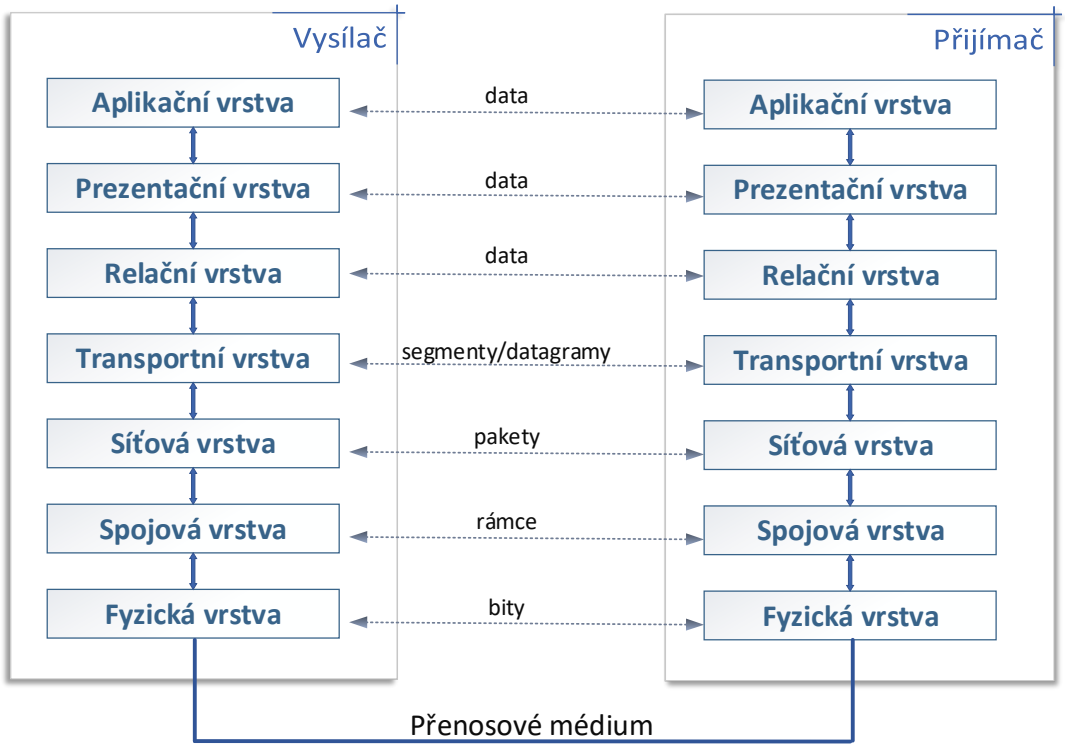
Obr. 1.2: Tvorba PDU jednotky [11].

jednotky PDU. Pokud PDU prochází od nejvyšší vrstvy po nejnižší, jsou metadata postupně nabalovány každou další průchozí vrstvou. Bezprostředně po předání jednotky PDU vyšší vrstvou vrstvě nižší se takto předaná data pro nižší vrstvu označují jako jednotka SDU (Service Data Unit). Z jednotky SDU daná vrstva vždy vytvoří jednotku PDU a poté ji zasílá dál. V opačném postupu průchodu jsou jednotlivé PDU postupně rozbalovány a dle režijních dat dané vrstvy je s daty individuálně zacházeno [5, 11].

Model ISO/OSI obsahuje 7 vrstev (obrázek 1.3):

- **Aplikační vrstva** - nejvyšší vrstva modelu, obsahující samotné aplikační procesy. Aplikace jsou ovládány samotnými uživateli a aplikační vrstva má tak za úkol poskytnout rozhraní mezi daty aplikací a nižšími vrstvami [5, 11].
- **Prezentační vrstva** - poslední vrstva, která umožňuje manipulaci přímo

s daty při možných transformací a úprav dat. Dále zajišťuje kompresi/dekompresi nebo šifrování/dešifrování [5],[11].



Obr. 1.3: Referenční model ISO/OSI [11].

- **Relační vrstva** - zajišťuje navázání relace a požadovaný přenos dat mezi konkrétními aplikacemi na straně odesílatele a příjemce. Relační vrstva může navázanou relaci přenosu dat mezi entitami kontrolovat a řídit [5].
- **Transportní vrstva** - dokáže protokolům vyšších vrstev zajistit vazbu mezi konkrétními aplikačními protokoly s požadovanou kvalitou služeb v závislosti na požadavku, zda je nutné poskytnout službu se spojením, nebo bez spojení. K adresaci aplikačních protokolů je využíváno portů. Dalšími funkcemi transportní vrstvy je segmentace na požadovanou velikost dat, kontrola toku dat a také kontrola chybových stavů [11].
- **Síťová vrstva** - má na starosti směrování a přenos dat mezi zařízeními ležící mimo lokální síť na základě logické síťové adresace a identifikace koncových bodů. Datové jednotky pracující na síťové vrstvě se označují jako pakety nebo datagramy [5, 11].
- **Spojová (linková) vrstva** - pracuje s rámci, které jsou vytvářeny z bitových toků. Rámec jako jediné PDU ze všech vrstev modelu ISO/OSI obsahuje i povinné zápatí. Spojová vrstva umožňuje adresaci mezi prvky připojenými

v lokální síti. Pomocí kontrolních vrstev LLC (Logical Link Control) a MAC (Media Access Control) dokáže spojová vrstva zajišťovat metody přístupu ke sdílenému médiu, či ovládání toku dat [5, 11].

- **Fyzická vrstva** - jedná se o nejnižší vrstvu modelu, jež je zodpovědná především za fyzický přenos výsledných bitů. Pro korektní přenos je nutné zajistit správné reprezentování bitů, vypořádat se přenosovými vlastnostmi pro dané přenosové médium, či rozhraní a také s korektní synchronizací přenosu [11].

Při shrnutí výše uvedených popisů všech 7 vrstev můžeme vrstvy rozdělit do 3 skupin. První skupinou jsou vrstvy orientované na podporu aplikaci (aplikační, prezentační a relační vrstva), druhou skupinou je samostatná vrstva transportní, která je zapotřebí pro určitou formu přizpůsobení. Poslední třetí skupinu nám uzavírají vrstvy orientované na přenos dat, do které patří fyzická, linková a síťová vrstva [5].

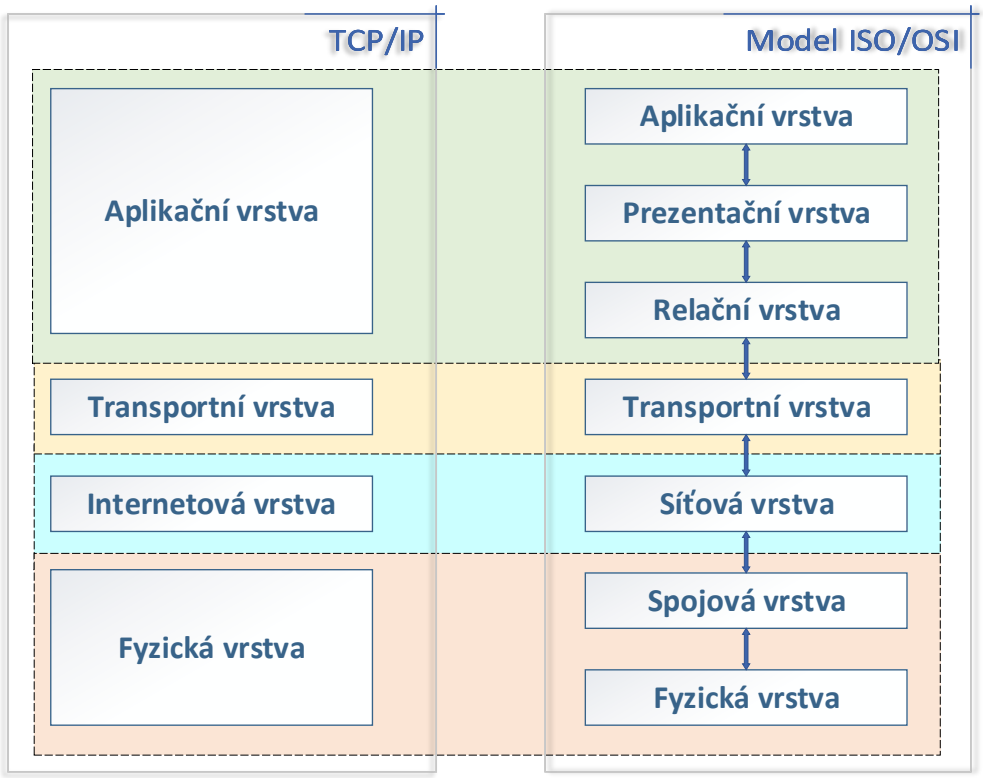
Přestože se autoři modelu ISO/OSI svou vymyšlenou koncepcí pokoušeli ujednotit a standardizovat principy komunikace síťových prostředků, postupem času se ukázalo, že navržený model není výhodné prakticky používat. Důvodů proč se model ISO/OSI neujal je hned několik. Především se jedná o představu, že veškeré odpovědnosti za přenos dat je kladena na síť, nikoliv na hostitelská zařízení. Další nevýhodou ISO/OSI se dají považovat také často opakované funkce na jednotlivých vrstvách, upřednostnění spojovaných přenosů. Navíc je z pohledu návrhu model ISO/OSI celkově zaměřen především na typ sítě WAN. Z těchto důvodů se model ISO/OSI používá v současnosti pouze jako teoretický model principů síťové komunikace [11].

1.4 Architektura TCP/IP

Síťový model, který se v dnešní době (na rozdíl od ISO/OSI) používá, je architektura TCP/IP, podle níž funguje i celosvětová síť Internet [11]. Často se používá i název rodina protokolů TCP/IP. Vlastnosti a směr, který TCP/IP nabízí se ukázal být velice výhodný pro plošné použití. Předpoklad menší složitosti architektury, nespojovaného charakteru, paketového přepojování s propracovanou možností vzájemného propojování dílčích sítí se v praxi prokázal jako velice úspěšné řešení. TCP/IP klade důraz především na koncová zařízení. Od přenosové sítě se požaduje pouze přenos dat, který by ale měl být maximálně efektivní [5, 12, 11].

Jak můžeme vidět na obrázku 1.4, který srovnává vrstvy obou zmíněných architektur, TCP/IP obsahuje pouze 4 vrstvy [3]:

- **Aplikační vrstva** - pracuje přímo s aplikačními procesy konkrétních programů. Protože TCP/IP neobsahuje relační a prezentační vrstvu musí se aplikace samy vypořádat s případnou potřebou navazování relací nebo k zajištění



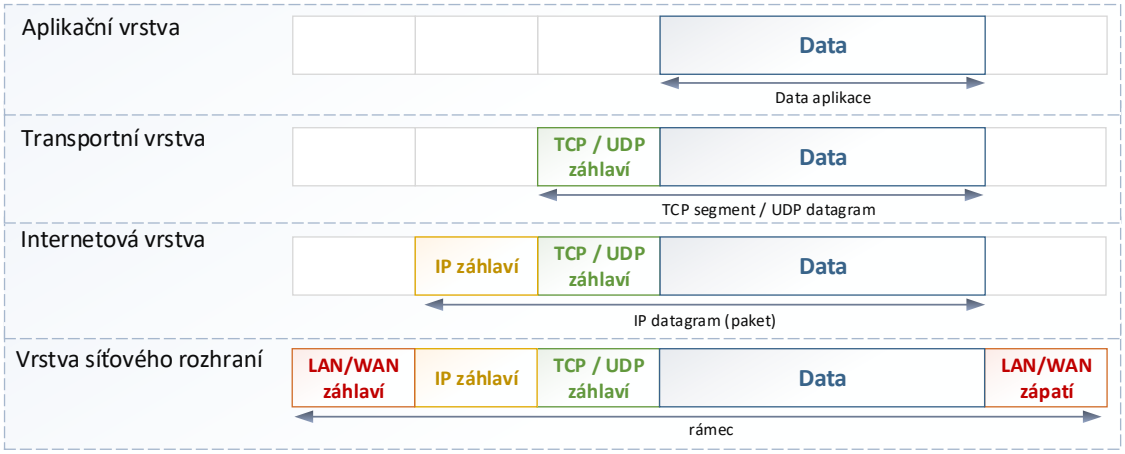
Obr. 1.4: Srovnání modelu TCP/IP a ISO/OSI [5].

komprese dat. Protokoly aplikační vrstvy musí taktéž umět přímo komunikovat s protokoly transportní vrstvy [5].

- **Transportní vrstva** - zajišťuje přenos dat přímo mezi aplikačními procesy, které se identifikují pomocí tzv. portů. Nejznámější protokoly transportní vrstvy jsou protokoly TCP (Transmission Control Protocol) a UDP (User Datagram Protocol). Pokud aplikace vyžaduji spojitý a spolehlivý přenos využijí protokol TCP. V případě, že není kladen důraz na spolehlivost přenosu, ale je vyžadován rychlý přenos krátkých zpráv, pak je k dispozici protokol UDP [5, 12].
- **Internetová (síťová) vrstva** - svými vlastnostmi odpovídá síťové vrstvě ISO/OSI. Rozdíl je však v poskytování výrazně jednoduší datagramové služby, která zaručuje nespolehlivý přenos. Na internetové vrstvě dochází také k fragmentaci, opětovnému sestavení a směrování do jiných sítí [5, 12].
- **Vrstva síťového rozhraní** - nejnižší vrstva zabývající se paketovým vysíláním, přijímáním a veškerým řízením přenosů. Dochází zde k zapouzdření IP datagramů do rámců, které jsou následně odesílány. Vrstva síťového rozhraní se přizpůsobuje v závislosti na druh připojené sítě [12].

Podobně jako u ISO/OSI architektura, TCP/IP používá postupné zapouzdření

na všech vrstvách. Graficky znázorněný postup zapouzdření je k vidění na obrázku 1.5. Data aplikačního protokolu při průchodu transportní vrstvou obdrží TCP, případně UDP záhlaví a jako výsledný segment postupuje dále na nižší internetovou vrstvu. Zde se ze segmentu se stává paketem tím, že je k jednotce SDU přidáno IP záhlaví. Dalším krokem je přidání LAN/WAN záhlaví a zápatí, které je závislé na typu síťového rozhraní. Tímto je vytvořený rámec, který je připraven k odeslání [5].



Obr. 1.5: Princip zapouzdření na vrstvách TCP/IP [5].

Nejznámější a nejpoužívanější protokoly architektury TCP/IP, se kterými se bude pracovat ve vytvořených laboratorních úlohách, budou podrobněji popsány v následující kapitole 2.

2 Protokoly architektury TCP/IP

Aby bylo možné zajistit korektní komunikaci dvou a více komunikujících stran tak, aby si mezi s sebou v sítí rozuměly, je nutné zajistit, aby se jejich komunikace řídila dle striktně definovaných pravidel [11]. Taková pravidla lze určit definováním daného protokolu, jež má za úkol určit co přesně má být danou komunikací sděleno, jakým způsobem a kdy bude konkrétní komunikace probíhat. Jako hlavní stavební prvky protokolů je možné považovat [5, 11]:

- Syntaxe - Zabývá se strukturou nebo formátem dat a to bez ohledu na jejich konkrétní význam. Syntaxe především ověřuje průchodnost kanálů, redundanci signálu a také způsoby daného kódování [5, 11].
- Sémantika - Na rozdíl od syntaxe, sémantika má za úkol zkoumat význam a obsah informace probíhajících dat daného protokolu. Získaná informace je využita pro případnou opravu chyb a pro řízení spolupráce mezi vrstvami protokolů [5, 11].
- Synchronizace - Zajišťuje korektní načasování komunikace protokolu. Především se jedná o počet opakování, kdy a jakou rychlostí mají být data odeslána nebo přijata [5, 11].

Síťové protokoly jsou definovány v tzv. RFC článcích, jež jsou spravovány skupinou specialistů, označovanou zkratkou IETF (Internet Engineering Task Force). Skupina IETF při vývoji a vydávání specifikací síťových protokolů spolupracuje také se skupinou IAB (Internet Architecture Board). V tomto případě se jedná o radu, která má za úkol záležitosti týkající se dohledu a celkového vývoje sítě Internet. Specifikaci používaných síťových protokolů tato skupina IETF publikuje na internetové stránky[1], kde jsou konkrétní dokumenty volně dostupné. Většina síťových protokolů bývá postupem času často upravovány a proto mnoho síťových protokolů můžeme nalézt hned v několika verzích. [11, 13]

V následující části práce budou představeny základní protokoly architektury TCP/IP, se kterými se bude nejvíce pracovat ve vypracovaných laboratorních úlohách. Z tohoto důvodu se v následujícím textu nenachází vybrané protokoly vrstvy síťového rozhraní.

2.1 Vybrané protokoly internetové vrstvy

Jak již bylo uvedeno v kapitole 1.4, hlavními úkoly internetové (síťové) vrstvy je logická adresace síťových prvků a funkce směrování mezi jednotlivými sítěmi. Směrování je zajištěno i mezi sítěmi, které jsou od sebe odlišné použitou technologií či

[1] https://www.ietf.org/standards/rfcs/

topologií. V architektuře TCP/IP internetová vrstva poskytuje nespojovanou datagramovou službu. Důraz je zde kladen na rychlost přenosu, při které však není zajištěna garance spolehlivosti korektního doručení. Základním protokolem internetové vrstvy je tzv. Internet protokol, zkráceně IP [5, 12].

2.1.1 IPv4

Internet Protocol verze 4 je jedním z hlavních komunikačních protokolů používaný od roku 1981 až dodnes [12]. Pro protokol IPv4 je typické zejména logické adresování síťových prvků pomocí 32 bitové IP adresy. Způsob takové adresace nabízí přes 4 miliardy (2^{32}) unikátních adres[2]. Vzhledem k současnému enormnímu nárůstu počítačových sítí a také vzrůstajícímu počtu různorodých komunikujících zařízení vyšlo najevo, že daný rozsah adres protokolu IPv4 je pro celosvětové použití nedostačující. Další nevýhodou IPv4 se dají považovat i chybějící prvky zabezpečení [11]. Tento problém se v rámci IPv4 sítí musí řešit pomocí jiných protokolů. Formát IPv4 záhlaví je k vidění na obrázku 2.1 [12].

Formát záhlaví IPv4 paketu

<table>
<tr><th>bity</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0</td><td colspan="4">Verze IP</td><td colspan="4">Délka záhlaví</td><td colspan="8">Typ služby</td><td colspan="16">Celková délka IP datagramu</td></tr>
<tr><td>32</td><td colspan="16">Identifikace IP datagramu</td><td colspan="3">Příznaky</td><td colspan="13">Posunutí fragmentace od počátku</td></tr>
<tr><td>64</td><td colspan="8">Doba života (TTL)</td><td colspan="8">Protokol vyšší vrstvy</td><td colspan="16">Kontrolní součet záhlaví</td></tr>
<tr><td>96</td><td colspan="32">IP adresa odesílatele</td></tr>
<tr><td>128</td><td colspan="32">IP adresa příjemce</td></tr>
<tr><td>160</td><td colspan="32">Volitelné položky záhlaví (0-320 bitů)</td></tr>
<tr><td>160 : 480</td><td colspan="32">Data</td></tr>
</table>

Obr. 2.1: Formát záhlaví IPv4 paketu [5].

Záhlaví obsahuje následující položky [11, 13]:

- Verze IP - v případě protokolu IPv4 je hodnota nastavena na 4 (binárně 0100).
- Délka záhlaví - udává počet 32 bitových slov (4B), které definuje možnou velikost záhlaví. Standardně je nastaveno na 5, což se rovná minimální velikosti záhlaví 20 B. Maximálně může záhlaví dosahovat 60 B, což se rovná počtu 15 [13].

[2]Ve skutečnosti je dostupných adres méně. Do rozsahu volných adres není možné počítat adresy rezervované pro speciální účely.

- Typ služby - pro speciálních služby aplikačních protokolů, jako jsou například služby přenosu hlasu či obrazu v reálném čase, je možné upravit možnosti směrování tak, aby byla poskytnuta požadovaná kvalita služby (QoS) [13].
- Celková délka IP datagramu - délka datagramu včetně záhlaví, která je udávána v bajtech. Minimum je velikost záhlaví bez dat (20 B) maximum je hodnota 65536 B. Velikost datagramů však bývá omezena MTU (Maximum Transmission Unit), tedy jednotkou, která definuje maximální velikost datagramu, v sítích Ethernet je tato hodnota nastavena standardně na 1500 B. IP datagramy tedy musí být upravovány na požadovanou maximální velikost - probíhá tzv. fragmentace [11].
- Identifikace IP datagramu - identifikace sloužící k rozpoznání přidružených fragmentů.
- Příznaky - jedná se o 3 bity které řídí případnou fragmentaci. První bit je nulový, druhý bit je označen jako DF a třetí MF. Bit MF označuje, že probíhající datagram je fragmentován a musí následovat jeho další část, zatím co bit DF identifikuje nefragmentovaný IP datagram [13].
- Posunutí fragmentace od počátku - pozice aktuálního datagramu vůči místu, kde začíná fragmentovaná zpráva.
- TTL (Time To Live) - hodnota zabraňující případnému zacyklení tím, že při každém průchodu směrovačem bývá ponížena o hodnotu 1. Jakmile TTL dosáhne hodnoty 0, je paket zahozen [11, 13].
- Protokol vyšší vrstvy - definují protokol vyšší vrstvy, kterému data patří.
- Kontrolní součet záhlaví - pokud se zjistí že součet záhlaví nesedí, tak dochází k zahození IP datagramu.
- IP adresa odesílatele - IPv4 identifikace síťového rozhraní, které IP datagram odeslalo.
- IP adresa cíle - IPv4 identifikace síťového rozhraní, který je adresátem IP datagramu.
- Volitelné položky záhlaví - většinou nevyužité možnosti rozšíření, které jsou nepovinné. Velikost volitelných položek může být maximálně 320 bitů (40 B).
- Data - zapouzdřená data předaná transportní službou [11].

IPv4 adresace

Pro identifikaci a odlišení síťových zařízení, protokol IPv4 využívá adresy o velikosti 32 bitů. Namísto ne příliš uživatelsky přívětivému zápisu v binární soustavě se IPv4 adresa nejčastěji udává v desítkové notaci. IPv4 adresa je tedy složena čtveřicí kladných celých čísel v rozsahu 0-255, které jsou označována jako oktety (oktet = 8 bitů) [13]. Jednotlivé oktety jsou mezi s sebou oddělené tečkami. Způsob

zápisu IPv4 adres bývá označován také jako tečkovaná desítková notace [11].

IPv4 adresy je možné rozdělit na dvě základní části. První část definuje síť, ve které se dané zařízení nachází a část druhá naopak konkrétní stanici. Hranici mezi těmito částmi určuje takzvaná maska sítě [13]. V binárním tvaru se v případě masky sítě jedná o posloupnost zleva jdoucích jedniček až do místa, kde se nachází zmíněná hranice mezi adresou sítě a adresou konkrétní stanice (hosta). Od této hranice je poté adresa konkrétního rozhraní definována posloupností nul. Masku sítě je možné zapisovat stejně jako IPv4 adresu v binárním tvaru, v tečkované desítkové notaci a navíc i pomocí prefixu. Hodnota prefixu je dána počtem bitů, určujících adresu sítě. Zapisován je za znakem lomeno (/) za přidruženou IPv4 adresou [11]. Ukázka možného zápisu IPv4 adres:

- IPv4 adresa (binárně): 11000000 10101000 00000001 00000101
- IPv4 adresa (tečkovaná desítková notace): 192.168.1.5
- Maska sítě (binárně): 11111111 11111111 11111111 00000000
- Maska sítě (tečkovaná desítková notace): 255.255.255.0
- IPv4 adresa (zápis pomocí prefixu: 192.168.1.5/24

Dle velikosti prefixu, byly IPv4 adresy rozděleny do tříd podle prvních bitů adresy. V závislosti na dané třídě byly definovány maximální počty adres určené pro koncové stanice a pro jednotlivé sítě. Rozdělení tříd se nachází v tabulce 2.1.

Tab. 2.1: Historické rozdělení IPv4 adres do tříd [11].

Třída	**1. oktet**	**Maska sítě**	**Možných sítí**	**Možných stanic**
A	0-127	255.0.0.0	128	11 677 214
B	128-191	255.255.0.0	16 384	65 534
C	192-223	255.255.255.0	2 097 152	254
D	224-239	Rozsah určený pro multicastové adresy		
E	240-250	Rozsah rezervních adres		

Toto řešení se však časem ukázalo jako neefektivní plýtvání již omezeného adresního rozsahu a proto byla zavedena tzv. beztřídní metoda CIDR (Classless Inter-Domain Routing), která dovoluje libovolnou velikost síťové masky (prefixu) v celkovém rozmezí 32 bitů. Délka prefixu tedy může být volena efektivně, dle požadavků konkrétních sítí. Metoda vytváření jednotlivých podsítí se označuje jako tzv. podsíťování (v anglickém překladu: subnetting) [11].

Speciální IPv4 rozsahy a adresy

V rámci IPv4 jsou definovány takzvané privátní rozsahy adres [13]:

- 10.0.0.0 – 10.255.255.255,

- 172.16.0.0 – 172.31.255.255,
- 192.168.0.0 – 192.168.255.255.

Tyto rozsahy jsou velmi často používány pro soukromé účely ve většině domácích či podnikových lokálních sítích. Uvedené privátní rozsahy jsou speciální tím, že se nesměrují a nemohou být použity jako veřejné. Aby tyto sítě mohly komunikovat s ostatními sítěmi je nutné na koncových směrovačích využít techniku překladu adres NAT (Network Address Translation) z privátní adresy na veřejnou, která je v rámci celosvětové sítě Internet unikátní [13]. Vedle zmíněných privátních rozsahů, stojí za zmínku i další speciální IPv4 rozsahy a adresy [11, 14]:

- 0.0.0.0/8 - adresa určená pro vlastní identifikaci konkrétních zařízení.
- 127.0.0.0/8 - lokální smyčka. Využívá se většinou pro lokální testování určitých aplikací na daných zařízeních. Vyslaný paket nikdy neopustí dané zařízení.
- 169.254.0.0/16 - rozsah adres, který se automaticky nastaví v případě selhání automatické konfigurace IP pomocí příslušného DHCP serveru.
- 224.0.0.0/4 - adresy rezervované pro vícesměrové vysílání (multicast).
- 255.255.255.255/32 - všesměrová adresa. Pokud je tato adresa nastavena jako cílová, je vyslaný paket rozeslán na všechna zařízení v dané síti. Jedná se vždy o poslední volnou adresu v konkrétní síti.

2.1.2 ICMPv4

Internet Control Message Protocol verze 4 je považován jako pomyslné servisní rozšíření protokolu IPv4. Jak bylo uvedeno v předešlé kapitole 2.1.1, protokol IPv4 poskytuje nespojované datagramové spojení. Může se tedy stát, že odeslaná data nedorazí do určeného cíle. Pro kompenzaci těchto nedostatků lze použít protokol ICMP [11]. Díky protokolu ICMP je umožněno pomocí servisních a provozních služeb získávat cenné informace o stavu sítě nebo o dostupnosti požadovaných adresátů a služeb [5].

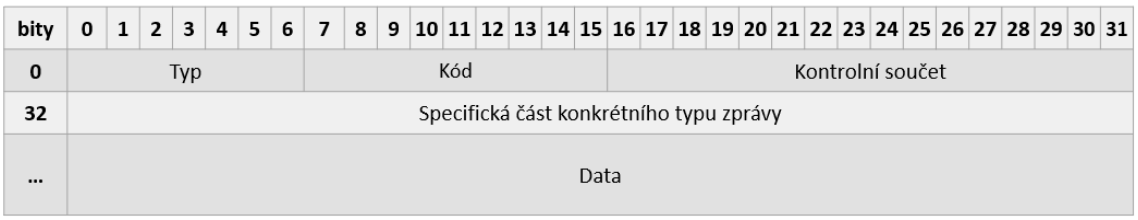
Záhlaví ICMPv4 zprávy

Jestliže je ICMPv4 rozšířením protokolu IPv4, tak je zřejmé, že zpráva protokolu ICMPv4 musí být součástí právě IPv4 datagramu. Záhlaví ICMPv4 zprávy je tedy zapouzdřené přímo za záhlavím datagramu IPv4 [5].

Velikost ICMPv4 záhlaví má velikost 64 bitů. Prvních 32 bitů záhlaví mají všechny ICMPv4 zprávy společné a další část se liší v závislosti na typu dané zprávy. Ve společné části záhlaví se nachází položky [11]:

- Typ - celočíselná hodnota definující typ zprávy.
- Kód - pole kód detailněji specifikuje konkrétní typ zprávy.
- Kontrolní součet - velikost kompletní ICMPv4 zprávy.

Poté následuje individuální záhlaví dle typu zprávy a samotná data. Záhlaví zprávy ICMPv4 reprezentuje obrázek 2.2.



Obr. 2.2: Záhlaví zprávy ICMPv4 [5]

Typy ICMPv4 zpráv

Typy zpráv je možné rozdělit do dvou kategorií. První kategorie zahrnuje zprávy s hlášením chybových stavů a druhá zprávy dotazovací. Typ zpráv nesoucí hlášení chybových stavů mají v datové části informace pro vyhledání původního paketu, zatím co druhý typ zpráv nese upřesňující informace požadovaných dotazů. Mezi nejznámější typy zpráv nesoucí hlášení chybových stavů (a příklady jejich rozšiřujících kódů) patří [11, 15]:

- Typ 3 - Destination unreachable: značí chybu v doručení IP datagramu.
  - Kód 0 - specifikuje nedostupnost sítě.
  - Kód 1 - specifikuje nedostupnost síťového rozhraní adresáta.
  - Kód 2 - specifikuje nedostupnost portu cílového protokolu.
  - Kód 6 - informuje o neznámé IPv4 adrese adresáta.
- Typ 4 - Source quench: informuje o potřebě snížení přenosové rychlosti, z důvodu, že směrovač nebo cílový prvek nestíhá přijímat IP datagramy.
- Typ 5 - Redirection: chyba ve směrovací tabulce odesílatele paketu.
- Typ 11 - Time exceeded: nedoručení z důvodu vypršení časového limitu [11, 15].
  - Kód 0 - Hodnota TTL byla ponížena na hodnotu 0 na směrovači a to před doručením paketu.
  - Kód 1 - Na cílové adrese došlo k vypršení časového limitu pro sestavení všech fragmentovaných zpráv.
- Typ 12 - Parameter problem: oznamuje chybu v záhlaví přijatého IP datagramu [11, 15].
  - Kód 0 - Chyba se nachází v parametru záhlaví.
  - Kód 1 - Parametr záhlaví nebyl nalezen.

Nejznámější typy zpráv z kategorie, sloužící k dotazování jsou [11, 15]:

- Typ 8 - Echo request: požadavek ověření dostupnosti adresáta.

- Typ 0 - Echo reply: odpověď na požadavek ověření adresáta.
- Typ 13 - Timestamp request: požadavek na zaslání časového razítka, pomocí výpočtu hodnoty RTT (Round Trip Time), sloužícího k synchronizaci.
- Typ 14 - Timestamp reply: odpověď na požadavek zaslání časového razítka.
- Typ 17 - Address Mask Request: požadavek o zaslání masky sítě.
- Typ 18 - Address Mask Reply: odpověď na požadavek o zaslání masky sítě.

Diagnostický nástroj PING

Zkratka PING je anglická zkratka slovního spojení Packet InterNet Groper [11]. PING je velmi využívaný diagnostický nástroj, používaný v případech, kdy je potřeba ověřit spojení či dostupnost síťového zařízení. Nástroj PING periodicky vysílá ICMP zprávy typu 8 (Echo request) a na základě odezvy obdržených odpovědí poskytuje informace o velikosti zpoždění a hodnotě TTL. K zjištění velikosti zpoždění je využíváno hodnoty RTT, přičemž výsledná hodnota je udávána v jednotkách milisekund. Po dokončení příkazu jsou obdržená data statisticky vyhodnocena [11].

Na obrázku 2.3 se nachází ukázka výstupu aplikace PING při zadání příkazu: ping 77.75.75.176 [3].

```
C:\Users\jciganek>ping 77.75.75.176

Pinging 77.75.75.176 with 32 bytes of data:
Reply from 77.75.75.176: bytes=32 time=8ms TTL=56
Reply from 77.75.75.176: bytes=32 time=8ms TTL=56
Reply from 77.75.75.176: bytes=32 time=7ms TTL=56
Reply from 77.75.75.176: bytes=32 time=7ms TTL=56

Ping statistics for 77.75.75.176:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 8ms, Average = 7ms
```

Obr. 2.3: Ukázka výpisu příkazu ping.

Na základě počtu přijatých a ztracených paketů je procentuálně vypočítána výsledná ztrátovost měření a dle naměřených zpoždění je uvedena minimální, maximální a průměrná hodnota zpoždění [11]. Nástroj PING se nachází ve většině operačních systémů a je spustitelný z příkazové řádky zadáním příkazu ping a zadáním příslušné IP adresy nebo kompletního doménového jména. Pokud je zadáno doménové jméno, tak příkaz automaticky zažádá o překlad IP adresy DNS server

[3] Jedná se o IP adresu internetové stránky <www.seznam.cz>

a příkaz je pak proveden na přeloženou IP adresu [11]. Princip nástroje PING je na všech operačních systémech stejný, avšak ve verzích pro OS Windows a OS Linux se liší volitelné parametry měření. Nástroj PING umožňuje upravovat poměrně velké množství parametrů měření. Pro ukázku budou uvedeny některé příklady volitelných parametrů nástroje PING ve verzi z OS Windows 10, které lze mezi sebou kombinovat [11]:

- -t - nepřetržité měření, které je ukončeno zadáním klávesy CTRL+C.
- -n <počet> - provede se zadaný počet měření.
- -S <IP_adresa> - pro měření nastaví zadanou IP adresu jako zdrojovou.
- -l <velikost> - nastaví velikost zasílaných zpráv.
- -w <čas> - nastavení maximální doby čekání na získání odpovědi. Zadává se v jednotkách milisekund.

Diagnostický nástroj TRACEROUTE/TRACERT

Dalším velmi používaným diagnostickým nástrojem pracujícím na protokolu ICMPv4, je nástroj TRACEROUTE [4]. TRACEROUTE je schopný, pomocí identifikace procházejících síťových uzlů zjistit přenosovou trasu odeslaného IP datagramu [11]. Pro zjištění přenosové trasy nástroj TRACEROUTE využívá získaných informací z příchozích odpovědí od prvků sítě, kterými zaslané pakety prochází [11]. Nejdříve je zaslaná tzv. sonda. V případě verze TRACERT, pro operační systém Windows, se jedná o ICMPv4 zprávy typu 8 (Echo request) s upraveným parametrem TTL, které zajistí, aby ihned na prvním procházejícím síťovém prvku vypršela platnost parametru TTL. Verze TRACEROUTE (pro OS Linux) využívá stejného principu upravení parametru TTL avšak ve výchozí konfiguraci využívá navázání komunikace pomocí protokolu UDP [11]. Prvek, u něhož dojde k vypršení hodnoty TTL, paket automaticky zahazuje a odpovídá zprávou typu 11 (Time exceeded) s kódem 0. V těle odpovědi se nachází hledaná IP adresa, která je zaznamenána. Spolu s IP adresou je zaznamenáno také případné doménové jméno prvku a výsledný čas od vyslání sondy po obdržení odpovědi. V základním nastavení se provádějí 3 měření, které jsou součástí konečného výpisu. Pokud není obdržena odpověď do stanoveného limitu (ve výchozí konfiguraci se jedná o 5 vteřin), je ve výpisu zobrazen znak *. Následuje stejný proces, ale nyní s navýšenou hodnotou parametru TTL o 1. Tento proces se opakuje do doby, dokud není dosaženo cílové adresy, nebo dokud není překročena hodnota maximálního počtu přeskoků TTL [5, 11, 16].

V dnešní době, kdy je kladen větší důraz na zabezpečení počítačových sítí, se může stát, že v některých sítích jsou ICMP zprávy blokovány bezpečnostními prvky.

[4] Pojmenování TRACEROUTE se využívá na linuxových operačních systémech. OS Windows používá název TRACERT.

Nástroj TRACEROUTE proto umožňuje navazovat spojení sond i pomocí protokolů transportní vrstvy TCP nebo UDP. Metodu navazování spojení je možné volit dle volitelných parametrů a tím se tak pokoušet o obejití stanovených bezpečnostních pravidel. V případě protokolu UDP se může využít metoda spojení na známé síťové porty (například port 53 protokolu DNS), nebo naopak na s vysokou pravděpodobností nepoužívané síťové porty. Při použití TCP spojení se velmi často využívá technika „napůl otevřeného spojení“, kdy nedochází ke kompletnímu navázání spojení [11].

Podobně jako nástroj PING, se nástroj TRACEROUTE (TRACERT) používá z příkazové řádky a je součástí většiny OS. Ukázka výpisu po zadání příkazu tracert z operačního systému Windows 10 se nachází na obrázku 2.4.

```
C:\Users\jciganek>tracert 77.75.75.176

Tracing route to www.seznam.cz [77.75.75.176]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.1.1 [192.168.1.1]
  2     1 ms     1 ms     1 ms  172.16.0.1 [172.16.0.1]
  3     *        *        *     Request timed out.
  4    10 ms    10 ms    10 ms  85.13.96.33
  5    10 ms    10 ms    10 ms  ons2.cd-t.cz [81.19.33.34]
  6     8 ms     8 ms     8 ms  nix2.seznam.cz [91.210.16.194]
  7     8 ms     8 ms     8 ms  n7k-ng-a-vdc-1-po1.seznam.cz [185.66.188.9]
  8     8 ms     8 ms     8 ms  n7k-ng-a-vdc-2-po3.seznam.cz [185.66.188.21]
  9     8 ms     8 ms     7 ms  www.seznam.cz [77.75.75.176]

Trace complete.
```

Obr. 2.4: Ukázka výpisu příkazu tracert.

Jediným povinným parametrem příkazu traceroute (tracert) je zadání cílového adresy. Cílovou adresu je možné zadat pomocí IP adresy nebo také pomocí doménového jmenného názvu. Nástroj TRACEROUTE obsahuje velké množství volitelných parametrů. Následuje ukázka často používaných volitelných parametrů linuxové verze TRACEROUTE [16]:

- -4, -6 - výběr mezi verzí pro IPv4 nebo IPv6. Bez zadání těchto parametrů je nastaveno IPv4.
- -I - určení metody navázání spojení sond pomocí ICMP.
- -T - určení metody navázání spojení sond pomocí TCP.
- -i <síťové_rozhraní> - specifikace síťového rozhraní, ze kterého jsou pakety vysílány.
- -w <čas> - specifikace délky času pro čekání na odpověď. Bez zadání parametru je automaticky nastaveno čekání 5 vteřin.

- -m <číslo> - specifikace maximální hodnoty TTL. Výchozí hodnota TTL je 30.
- -l <velikost> - nastaví velikost zasílaných zpráv.
- -p <číslo_portu> - nastavení cílového portu.
- -q <počet> - specifikace počtu zaslaných sond a tím i počtu měření zpoždění. Pokud není zadáno, je ve výchozím nastavení nastavena hodnota 3.

2.1.3 IGMP

Protokol IGMP (Internet Group Management Protocol) [11] lze považovat jako doplněk protokolu IPv4, který umožňuje použití skupinového (vícesměrového) vysílání v rámci IPv4 sítě. Hlavní úlohou uvedeného protokolu je správa multicastových skupin. Koncové stanice, tak pomocí IGMP protokolu, mohou požádat o přístup do skupinové adresy svůj přidružený směrovač [11]. Tento směrovač je pak na základě IGMP zpráv informován, kterému zařízení přísluší adresa přijatého skupinového vysílání. Protokol IGMP existuje ve 3 variantách a to konkrétně IGMPv1, IGMPv2 a IGMPv3 [11, 13].

IGMPv1

Základní verze protokolu, jež má k dispozici pouze dva typy zpráv [13]:

- Membership Query - zpráva, pomocí které zjišťuje směrovač od lokálních stanic jejich žádosti o členství skupinových adres. Tato zpráva je směrovači zasílána periodicky (výchozí hodnota je nastavena na 60 vteřin) na skupinovou adresu 224.0.0.1 [13].
- Membership Report - odpověď na předchozí zprávu, kde lokální stanice posílají adresu skupinového vysílání, ke které se chtějí připojit. Aby nedocházelo k zahlcení v případě odpovědí většího počtu lokálních stanic, bývá tato zpráva odesílána v náhodně zvoleném časovém intervalu [13].

Pokud stanice během určitého časového intervalu neinformuje zprávou Membership Report o své žádosti pro danou skupinovou adresu, dochází k zastavení skupinového vysílání pro tuto stanici [11].

IGMPv2

Druhá verze protokolu IGMP, obsahující 4 typy zpráv [11, 17]:

- Membership Query - podobná zpráva jako u předchozí verze, zasílaná na adresu 224.0.0.1 periodicky. Tentokrát je výchozí hodnota nastavena na 125 vteřin. Existují dva podtypy této zprávy [11, 17]:
  - General - dotaz na všechny skupiny, ke kterým jsou připojené lokální stanice.

  - Group-specific - dotaz zda existuje zájemce (odběratel) pro konkrétní specifikovanou skupinu.
- Membership Report verze 2 - odpověď na dotaz protokolu IGMPv2, na kterou odpovídá pouze jeden člen skupiny.
- Membership Report verze 1 - zpětně kompatibilní odpověď na dotaz protokolu IGMPv1.
- Leave Group - zpráva zasílána na adresu 224.0.0.2, pomocí které lokální stanice informuje o ukončení vysílaní a tím i o zrušení členství pro vybranou skupinovou adresu.

IGMPv2 [17] vylepšuje předchozí protokol především v efektivitě správy skupin multicastového vysílání. V případě většího počtu směrovačů posílá na rozdíl od předchozí verze zprávu Membership Query pouze jeden směrovač. Další vylepšení spočívá v tom, že lokální stanice nově umí poslat zprávu, jež informuje o ukončení vysílání pro danou skupinu. Upravena je také hlavička protokolu, kde se nově vyskytuje pole určující časový limit, během kterého musí lokální stanice odpovědět na výzvu Membership Query [13, 17].

IGMPv3

Nejnovější třetí verze protokolu IGMP rozšiřuje především možnosti přístupu do vybraných skupin, ze kterých je stanicím umožněna filtrace a přímý výběr jednotlivých zdrojů vysílání [18]. Není tedy nutné přijímat data od všech zdrojů dané multicastové skupiny, ale nyní si je možné z vysílacích zdrojů vybírat. Hlavička zpráv protokolu IGMPv3 navíc obsahuje jednak nová pole pracující s synchronizací časovačů, ale také počet a výčet všech zdrojových adres pro danou skupinovou adresu [13, 18].

Na základě výše uvedených nových možností jsou rozšířeny i dotazové zprávy typu Membership Query na 3 nové podtypy [18]:

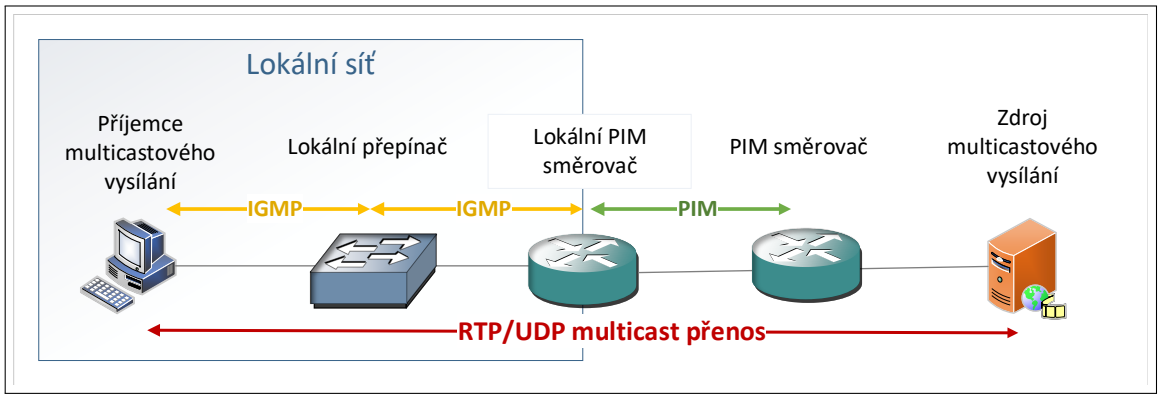
- Group-and-source-specific - dotaz na příjemce skupiny a konkrétně specifikovaných zdrojů. Tyto dotazy se odesílají na příslušnou skupinovou adresu.
- Group-specific - dotaz zaměřený pouze na konkrétní skupinu. Tyto dotazy se i v tomto případě odesílají na příslušnou skupinovou adresu.
- General - dotaz celkový stav žádostí o skupinové vysílání. Tato zpráva je zasílaná na adresu 224.0.0.1.

Modifikaci se dočkala i zpráva typu Membership Report, která je zasílána oproti předchozím verzím na adresu 224.0.0.22. Pomocí dvou vyhrazených módů Exclude a Include klientské stanice stanice informují přidružený směrovač o požadovaném přijmu vysílání multicastových skupin, případně vybraných konkrétních zdrojů vysílání. Výběr z jednotlivých zdrojů se provede pomocí módu Include s definovanými zdroji vysílání, zatímco módem Exclude jsou automaticky přiřazeny všechny zdroje.

Ve variantě IGMPv3 je pomocí zpráv Membership Report daný směrovač informován taky o ukončených skupinových vysílání. Pokud je zaslána zpráva Membership Report v módu Include kde není specifikovaný žádný zdroj, tak je pro konkrétní stanici ukončeno vysílání dané skupiny. Stejně jako protokol IGMPv2 je zpětně kompatibilní s první verzí, tak i protokol IGMPv3 je kompatibilní se všemi předchozími verzemi [13, 18].

2.1.4 PIM

Protokol PIM (Protocol Independent Multicast) patří do skupiny protokolů, které mají na starosti směrování multicastového přenosu mezi směrovači [11]. V součinnosti s protokolem IGMP je možné pomocí protokolu PIM v IPv4 sítí sestavit a provozovat kompletní vícesměrové vysílání [11, 13]. Ukázka zmíněné součinnosti protokolu PIM a IGMP je zobrazeno na obrázku 2.5.



Obr. 2.5: Součinnost protokolu PIM a IGMP v IPv4 síti [13].

Přestože je protokol PIM považován jako protokol směrovací, tak si ve skutečnosti nevede vlastní směrovací tabulku. Směrovací informace odebírá od libovolného směrovacího protokolů unicastového přenosu. Protokol PIM, tak může spolupracovat s protokoly OSPF (Open Shortest Path First), RIP (Routing Information Protocol) a podobně. Protokol PIM lze použít v různých variantách. Na základě zvolené varianty protokolu PIM jsou sestavovány distribuční stromy různými způsoby. Základní varianty protokolu PIM jsou [13, 19, 20]:

- Dense Mode (DM),
- Sparse Mode (SM),
- Sparse-Dense Mode (SDM),
- Rozšiřující varianty:
  - Bidirectional PIM (BIDIR-PIM),
  - Source-specific Multicast (SSM).

Dense Mode (DM)

Varianta směrování protokolu, která je výhodná zejména u takových sítí, kde se předpokládá, že téměř všichni účastníci budou přijímat skupinové vysílání. Mechanismus varianty Dense Mode na začátku vysílání rozesílá (zaplavuje) provoz mezi všemi definovanými PIM směrovači [19]. Pokud se stane, že určitý směrovač nemá připojené žádné zájemce o vysílání, tak tento směrovač odesílá zprávu pro potlačení daného skupinového vysílání. Potlačující zpráva (prune) má platnost 180 vteřin a po vypršení tohoto intervalu je znovu odesílána v případě, že daný směrovač stále nemá zájem o skupinové vysíláni [19]. Naopak směrovače, které mají platné zájemce o připojení, pomocí mechanismu RPF (Reverse Path Forwarding) určují optimální trasu vysílání (strom nejkratších cest) a ostatní pakety, jež přichází mimo optimální trasu zahazují [19].

Sparse Mode (SM)

Na rozdíl od předchozí varianty se vytváří tzv. sdílené stromy za pomoci speciálně definovaného směrovače, označovaného jako RP (Rendezvous Point). Tento směrovač se stává kořenem sdíleného stromu směrovačů, přes který je řízeno veškeré skupinové vysílání. Ostatní směrovače, které jsou vybrány jako směrovače multicastového vysílání pro jednotlivé lokální sítě jsou označovány jako DR (Designed Router) [20].

Princip Sparse Modu je koncipován tak, aby bylo skupinové vysílání zasíláno primárně pouze zájemcům o vysílání dané skupiny [20]. Zájemci o skupinové vysílání se hlásí o daný příjem u RP opakovanými zprávami Join skrze routery DR [20]. V případě, že již zájem nemají, informují o opuštění skupiny zprávou Prune. Ve fázi, kdy RP přijme žádosti příjemců, jsou vysílána data od zdroje vysílání nejdříve zapouzdřena a poté od svého přidruženého DR směrovače (pomocí zpráv Register) posílána na unicastovou adresu RP. RP data rozbalí a skrze vytvořený distribuční sdílený strom zasílá zvolenou cestou přes příslušné DR na přihlášené zájemce [20]. Po ukončení registrace zdroje (Register-Stop zpráva) jsou již od zdroje dat přeposílána nativní data přes RP přímo ke příjemci. Varianta Sparse Mode však za určitých podmínek (například když dojde k vytížení RP směrovače) umožňuje posílat data přímo od zdroje nejkratší cestou přímo k cílovému příjemci [20].

Sparse-Dense Mode (SDM)

Kombinace obou předchozích variant, jež je používána například na zařízeních Cisco. Pokud není žádný směrovač v sítí definován jako RP, funguje skupinové směrování na principu Dense Mode. V opačném případě je směrování řízeno dle principu Sparse Mode.

Rozšiřující varianty PIM

- Bidirectional PIM (BIDIR-PIM) - rozšiřující varianta typu SM, která umožňuje vytvoření jednoho vícesměrového distribučního stromu, kde je možné více uživatelů připoj k většímu počtu zdrojů [11, 13].
- Source-specific multicast (SSM) - jak už název napovídá, tato varianta zajišťuje skupinové vysílání pouze pro jeden konkrétní zdroj. Varianta SSM spolupracuje s protokolem IGMPv3 a přestože se jedná o upravenou variantu SM není zde pro směrování využito směrovače RP a příjemci se musí registrovat přímo u zdroje vysílání [11, 13].

Typy zpráv protokolu PIM

Protokol PIM ke svému korektnímu fungování používá několik typu zpráv, jež jsou většinou zasílány na adresu 224.0.0.13, která definuje všechny směrovače, kde je nakonfigurovaný protokol PIM [19, 20]. Základní zprávy protokolu PIM jsou:

- Hello - slouží k detekci všech směrovačů, kde je definován a nastaven protokol PIM. Zasílaná periodicky, výchozí interval opakování je 30 vteřin [19].
- Register (pouze u SM) - zprávu odesílá PIM směrovač, ke kterému je připojený zdroj vysílání. Cílem je unicastová adresa RP směrovače, který je touto zprávou (zapouzdřenou do unicastového vysílání) informován o aktivním vysíláním zdroje multicastových dat [20].
- Register-Stop (pouze u SM) - reakce RP směrovače na přijatou zprávu Register, kterou je DR směrovač zdroje informován o úspěšné registraci. Tímto DR směrovač zdroje ukončí posílání zapouzdřených dat a vytvořeným zdrojovým stromem posílá nezapouzdřená nativní data k RP, který je poté dál rozesílá k příjemci [20].
- Join/Prune - zpráva která ve svém těle, obsahuje seznamy zdrojů multicastového vysílání, u kterých dané směrovače chtějí pokračovat v připojení (Join), nebo o jejich vysílání již nemají zájem (Prune).
- Bootstrap (pouze u SM) - využívá mechanismy automatického delegování RP směrovače v rámci dané sítě [20].
- Assert - detekuje a zabraňuje duplicitě vysílaných tras. Děje se tak především u varianty DM, kdy na směrovač, který žádá o přijetí do skupiny dorazí vysílání ze 2 či více směru. Cílový směrovač si poté vybere vítězný směrovač, jehož zvolí na základě porovnání parametrů použitého unicastového směrovacího protokolu (Metric Preference), metriky a IP adresy [19].
- Graft (pouze u DM) - podobná zprávě Join s tím rozdílem, že tato zpráva slouží k opětovnému navázání spojení, které bylo v minulosti již navázáno a později přerušeno. Využívá se pouze ve variantě Dense Mode a zasílána je

na unicastovou adresu směrovačů [19].

- Graft-Ack (pouze u DM) - zpráva, která potvrzuje přijetí zprávy Graft, jež má využití pouze u varianty Dense Mode [19].
- Candidate-RP-Advertisement (pouze u SM) - směrovač touto zprávou ohlašuje zájem o kandidaturu na RP při mechanizmu automatického volení RP směrovače [20].

2.2 Vybrané protokoly transportní vrstvy

Transportní vrstva se v rámci modelu TCP/IP nachází na pomyslném rozhraní mezi nižšími vrstvami, které zajišťují komunikaci mezi vzdálenými stanicemi a vrstvou aplikační [11]. Proto transportní vrstva pracuje již s konkrétními procesy aplikací. Z tohoto důvodu musí být protokoly transportní vrstvy schopny zajistit korektní doručení dat jednotlivým aplikacím a tím pádem tak určitým způsobem jednotlivé procesy aplikací od sebe rozlišit [11]. K rozlišení jednotlivých procesů slouží 16-ti bitové čísla, nazývána jako porty. Port může být zdrojový, nebo cílový. V případě zdrojového portu se jedná o lokální proces, který zahajuje komunikaci se vzdáleným procesem, jež je reprezentován portem cílovým. Porty se dělí do tří hlavních skupin [5, 11]:

- Známé - číselný rozsah portů: 0 – 1023. Často používané a známé aplikace.
- Registrované - číselný rozsah portů: 1024 – 49151. Méně používané aplikace, které jsou registrované organizací IANA.
- Soukromé a dynamické - číselný rozsah portů: 49152 – 65535. Dynamicky přiřazované porty, které nejsou fixně spojovány s určitou aplikací a porty, které jsou vyhrazené jako soukromé.

Příklady známých portů nejpoužívanějších aplikačních protokolů se nacházejí v tabulce 2.2. Kombinaci IP adresy a daného portu nazýváme socket.

Tab. 2.2: Porty vybraných aplikačních protokolů [11].

Aplikační protokol	**Číslo portu**	**Transportní protokol**
FTP	20 / 21	TCP
TFTP	69	UDP
DNS	53	TCP/UDP
DHCP	67 / 68	UDP
HTTP	80	TCP
HTTPS	443	TCP/UDP

Hlavním úkolem transportních protokolů je poskytovat požadovanou službu konkrétním protokolům vyšší aplikační vrstvy [11]. Požadavky na služby transportních

protokolů se mohou lišit zejména v požadované spolehlivosti přenosových služeb, nebo potřeby spojovaného či nespojovaného přenosu, které jsou odvozeny například dle požadované rychlosti předání dat. Dle požadovaných vlastností je pak konkrétním aplikačním protokolům vybrán vhodný protokol transportní vrstvy [5, 11]. Nedílnou součástí práce transportních protokolů je také segmentace dat v případech, že aplikace přenáší velké množství dat, které je nutné rozdělit na několik menší oddílů. Před jednotlivé oddíly je poté přidáno identifikační záhlaví transportního protokolu a takto připravené jednotky jsou předány vrstvě síťové k následnému směrování. Provedení segmentace se však liší v závislosti na použitém transportním protokolu [5].

Nejpoužívanější protokoly transportní vrstvy jsou protokoly UDP a TCP. Příkladem dalších protokolů transportní vrstvy, které však nejsou tak často využívány, jsou protokoly SCTP (Stream Control Transmission Protocol) nebo protokol RTP (Real-time Transport Protocol) [3, 5].

2.2.1 UDP

Protokol UDP (User Datagram Protocol) [21] nabízí nespolehlivý a nespojovaný přenos. Jedná se o jednoduchý protokol, který nemá zakomponované žádné mechanismy řízení toku dat, kontrolu zahlcení nebo řízení chybových stavů [21]. Využití protokolu UDP dává smysl především v případech, kdy je požadován rychlý přenos krátkých zpráv, u kterých je akceptovatelné občasné selhání spojení. Typickým příkladem vyšších protokolů, kterým vyhovují vlastnosti UDP protokolu, jsou protokoly přenášející obraz a zvuk v reálném čase. Základní jednotkou, se kterou protokol UDP pracuje se nazývá datagram [11].

Záhlaví datagramu UDP

O složitosti protokolu UDP vypovídá i jednoduché záhlaví (viz obrázek 2.6). Záhlaví datagramu UDP obsahuje následující položky [11, 21]:

bity	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Zdrojový port																Cílový port															
32	Celková délka																Kontrolní součet															
64	Data																															

Obr. 2.6: Záhlaví datagramu UDP [21].

- Zdrojový port: celočíselný identifikátor zdrojového portu.
- Cílový port: celočíselný identifikátor cílového portu.

- Celková délka: délka celého UDP datagramu udávána v bytech. Může nabývat hodnot od 8 bajtů do 65535 bajtů.
- Kontrolní součet: nepovinná položka, která je v případě potřeby počítána z tzv. pseudo-hlavičky a vlastního UDP datagramu. Pseudo-hlavička je získána z dat aplikační vrstvy a obsahuje IP adresu odesílatele a příjemce. Hodnota kontrolního součtu slouží ke kontrole integrity přenosu.

2.2.2 TCP

TCP (Transmission Control Protocol) protokol poskytuje aplikačním protokolům spojovaný a spolehlivý charakter přenosu aplikačních dat [22]. Spolehlivost zaručují zakomponované mechanismy řízení toku dat, jež jsou schopné regulovat velikost přenášených dat tak, aby nedocházelo zahlcení přenosového média a tím pádem ke ztrátě, či zahození přenášených dat [11]. Rozdíl ve složitosti protokolu TCP, v porovnání s protokolem UDP, je patrný dle rozsáhlejšího záhlaví, které je na obrázku 2.7. Základní jednotka, se kterou protokol TCP pracuje se nazývá segment. Před začátkem přenosu dat protokol TCP nejdříve naváže komunikaci (pomocí tzv. three-way handshake) a po dokončení přenosu dochází k ukončení spojení. Pro správný průběh komunikace protokol TCP využívá příznakových bitů, stejně tak se využívá i číslování odeslaných a potvrzených bajtů či kontrolních součtů [11, 22].

Záhlaví segmentu TCP

<table>
<tr><th>bity</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0</td><td colspan="16">Zdrojový port</td><td colspan="16">Cílový port</td></tr>
<tr><td>32</td><td colspan="32">Pořadové číslo odesílaného bajtu</td></tr>
<tr><td>64</td><td colspan="32">Pořadové číslo potvrzovaného bajtu</td></tr>
<tr><td>96</td><td colspan="4">Délka záhlaví</td><td colspan="6">Rezerva</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="16">Délka okna</td></tr>
<tr><td>128</td><td colspan="16">Kontrolní součet</td><td colspan="16">Ukazatel naléhavých dat</td></tr>
<tr><td>160</td><td colspan="32">Volitelné položky</td></tr>
<tr><td>192</td><td colspan="24">Volitelné položky (pokračování)</td><td colspan="8">Výplňkové bity do 32 bitů</td></tr>
<tr><td>224</td><td colspan="32">Data</td></tr>
</table>

Obr. 2.7: Záhlaví segmentu TCP [22].

Záhlaví protokolu TCP obsahuje [11, 22]:

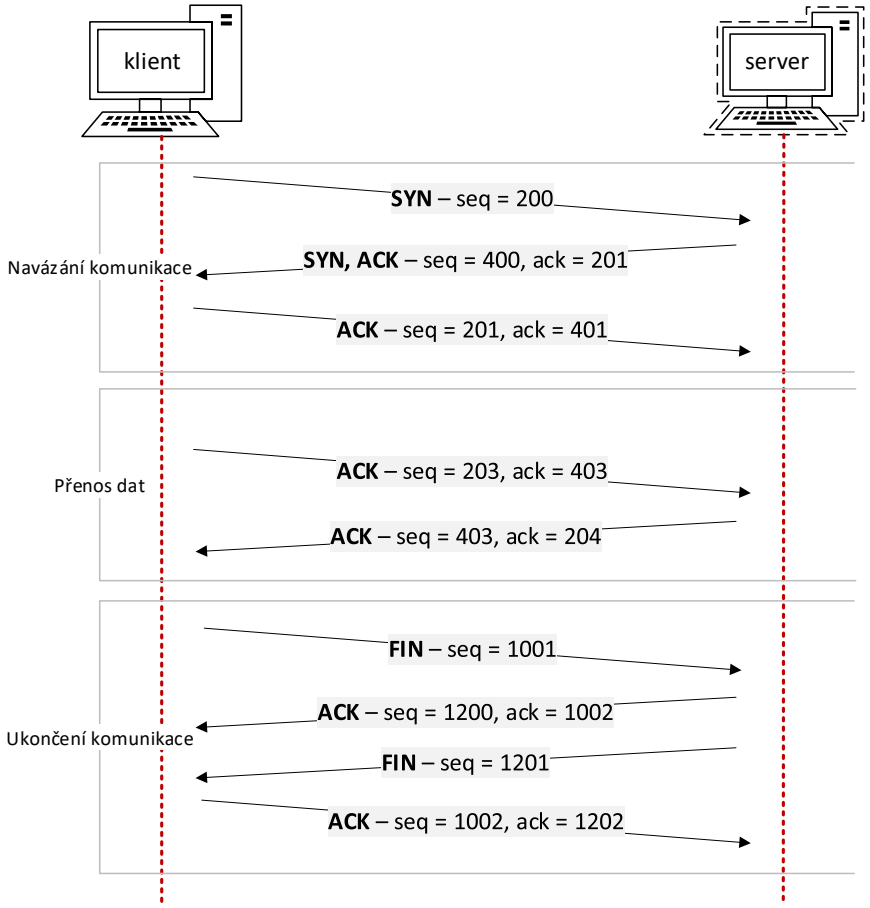
- Zdrojový port - celočíselný identifikátor zdrojového portu.
- Cílový port - celočíselný identifikátor cílového portu.

- Pořadové číslo odesílaného bajtu - číselný identifikátor prvního bajtu každého TCP segmentu, zajišťující identifikaci správného pořadí segmentů (seq - sequence number) [5].
- Pořadové číslo potvrzovaného bajtu - celočíselná hodnota prvního bajtu očekávaného segmentu. Pomocí pořadových čísel odeslaných a potvrzovacích bajtů je zaručena kontrola správnosti pořadí odeslaných segmentů. Hodnota pořadového čísla potvrzovaného bajtu (označována jako ack - acknowledgment number) je vždy hodnota, která je o 1 větší, než hodnota pořadového čísla přijatého bajtu [5].
- Délka záhlaví - položka, která udává celkovou velikost záhlaví. Zadává se počtem 32 bitových bloků, neboli počtem řádků záhlaví. Velikost TCP segmentu se pohybuje v rozmezí 20-60 bajtů. V přepočtu se tedy hodnota pole délky záhlaví udává v rozmezí 5-15 (32 bitových bloků)
- Rezerva - pole 6 bitů, které se v současnosti nevyužívá a je rezervováno pro případné budoucí využití.
- Příznaky - 6 důležitých příznakových bitů, sloužící ke korektnímu řízení komunikace [11, 22]:
  - URG - indikuje segmenty s naléhavými daty.
  - ACK - potvrzuje správné pořadí přijatých segmentů na základě korektního průběhu komunikace dle hodnot pořadového čísla potvrzovacích bajtů (ack).
  - PSH - označuje segmenty, nesoucí přímo aplikační data. Je očekáváno přednostní předání.
  - RST - příznak znamenající odmítnutí spojení.
  - SYN - indikuje navázání komunikace a tím i nastavení počáteční hodnoty pořadového čísla odesílaného bajtu (seq).
  - FIN - odesílatel tímto příznakem ukončuje navázané spojení.
- Délka okna - hodnota definující velikost dat, které je možné odeslat aniž by muselo dojít k jejich potvrzení příjemcem.
- Kontrolní součet - podobně jako u protokolu TCP slouží k zajištění integrity přenosu. Na rozdíl od protokolu UDP, je v případě TCP, kontrolní výpočet povinný. Počítán je ze záhlaví a dat segmentů doplněnými informacemi ze záhlaví IP datagramů [5].
- Ukazatel naléhavých dat - v případě segmentů obsahujících naléhavá data je v této položce definován ukazatel, který v daném segmentu značí konec naléhavých dat.
- Volitelné položky - nepovinné pole, které může dosahovat velikosti maximálně 40 bajtů. Volitelné položky mohou sloužit k poskytnutí nepovinných rozšiřujících informací jako jsou například časová razítka, nebo informace o požadované

velikosti segmentů. Vždy platí, že velikost volitelných položek musí být dělitelná číslem 4 [11, 22].

Průběh komunikace protokolu TCP

Aby protokol TCP dosáhl slibované spolehlivosti přenosu, využívá mechanizmu tzv. kladného potvrzování odeslaných segmentů. Aby nedocházelo k výraznějšímu zpomalení přenosů způsobených postupnou kontrolou každého odeslaného segmentu, využívá se metody „klouzavého okna“. Tato metoda s pomocí vyrovnávací paměti dovolí odesílat více segmentů za sebou bez nutnosti čekaní na jejich bezprostřední potvrzení. Pokud do vypršení limitu nedojde ke kladnému potvrzení segmentů, je vyžádáno opětovné odeslání chybně přenesených dat [5].



Obr. 2.8: Hlavní fáze komunikace protokolu TCP [22].

Na obrázku 2.8 jsou vyobrazené hlavní fáze komunikace protokolu TCP avšak ve zjednodušené podobě, kdy je očekávané postupné potvrzování jednotlivých segmentů. Mezi hlavní fáze průběhu TCP komunikace patří [5, 11]:

1. Navázání spojení - Pomocí 3 kroků probíhá navázání komunikace protokolu TCP. Nejdříve je odeslán segment s příznakem SYN, který je zaslán spolu s náhodně vygenerovaným pořadovým číslem odeslaného bajtu (seq), které je označováno jako číslo ISN (Initial Sequence Number). Druhým krokem je potvrzení přijetí předešlého segmentu příznakem ACK (za předpokladu, že je v pořádku pořadová hodnota očekávaného bajtu). Druhý krok zahrnuje také inicializaci spojení z druhého směru příznakem SYN, opět s vygenerovaným číslem (seq). Finální krok je potvrzení očekávaného pořadového bajtu žádosti SYN z druhého kroku.
2. Průběh spojení - Probíhá vzájemné potvrzování pořadových čísel odeslaných a očekávaných bajtů jednotlivých segmentů.
3. Ukončení spojení - využívá se příznaku FIN a jejich následného potvrzení ACK. Příznak FIN musí být odeslán z obou stran spojení a stejně tak musí být oboustranně potvrzený příznakem ACK.

Pro protokol TCP najdou využití především aplikace, které si za žádných okolností nemohou dovolit jakoukoliv ztrátu dat při přenosu. TCP protokol využívají aplikační protokoly jako například protokol FTP, HTTP a SMTP [5, 11].

2.3 Vybrané protokoly aplikační vrstvy

Protokoly aplikační vrstvy tvoří procesy, které poskytují službu přímo koncovým uživatelů. Aplikační vrstva má za důležitý úkol k zajištění korektní komunikace zprostředkovat spojení mezi daty aplikací a nižší transportní vrstvou. V rámci architektury TCP/IP drtivá většina aplikačních protokolů funguje na principu klient-server, jehož princip byl popsán již v kapitole 1.1.3 [9, 11].

2.3.1 DNS

DNS (Domain Name System) protokol umožňuje pomocí hierarchicky organizovaných DNS serverů překládat IP adresy síťových prvků pro uživatele mnohem lépe zapamatovatelná doménová jména [11, 25].

DNS protokol je v seznamu aplikačních portů možné nalézt pod číslem 53 [11]. Transportní protokol může být v případě DNS použit UDP i TCP v závislosti na velikosti přenášených dat. V případě běžných DNS dotazů, kdy je velikost DNS zprávy menší než nastavená hodnota MTU (Maximum Transmission Unit) je použit transportní protokol UDP, který protokolu DNS vyhovuje více, především díky menší režii přenosu a tím odpovídajících rychlostí vyřízení překladu. V případě větších DNS zpráv, než je stanovený tento limit, je použit spolehlivý protokol TCP [5].

Z historického hlediska specifikace DNS protokolu byl limit pro použiti UDP protokolu stanovený na hraniční hodnotu DNS zprávy 512 bytů [11].

Typy DNS záznamů

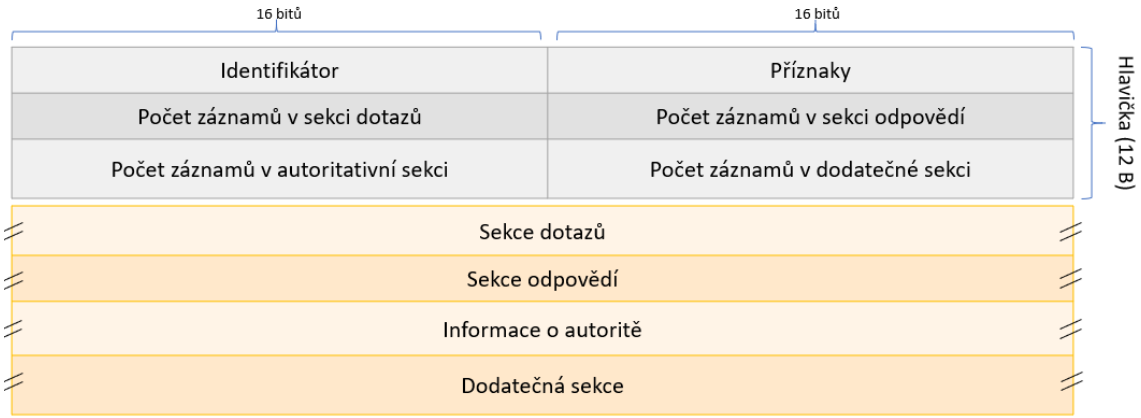
Záznamy, které jsou využívány protokolem DNS se nazývají zdrojové věty RR (Resource Records). Věty RR obsahují veškeré informace potřebné ke správnému fungování protokolu DNS. Formát RR věty se skládá z částí [11, 25]:

- Doménové jméno - pole s proměnnou délkou nesoucí doménové jméno prvku nebo subdomény.
- Typ - mezi základní typy vět RR lze uvést [11, 25]:
  - 1 - A - obsahuje konkrétní IPv4 adresu.
  - 2 - NS (Name server) - obsahuje doménové jméno autoritativního serveru dané sítě.
  - 5 - CNAME (Canonical name) - rozšíření umožňující použití dalších doménových jmen pro jeden konkrétní prvek.
  - 11 - WKS (Well-known services) - definuje služby, které poskytuje daný prvek.
  - 12 - PTR (Pointer) - určený pro reverzní překlad doménových jmen.
  - 13 - HINFO (Host information) - obsahuje základní informace o daném prvku.
  - 15 - MX (Mail) - nese jméno emailového serveru pro danou doménu.
  - 16 - TXT (Text string) - řetězec obsahující volitelný doplňkový text.
  - 28 - AAAA - obsahuje konkrétní adresu protokolu IPv6.
- Třída - definuje specifickou třídu vět.
- Doba života (TTL) - pole 32 bitů udávající dobu, po kterou je RR věta platná v dočasných pamětí DNS resolverů.
- Délka dat - pole o velikosti 16 bitů, které uvádí velikost pole pro data.
- Datová část - samotná data DNS protokolu, která jsou specifická v závislosti na daném typu a třídě.

Formát zprávy DNS protokolu

Na obrázku 2.9 se nachází schéma kompletní DNS zprávy, která se skládá z následujících částí [11, 25]:

- Hlavička (header) - povinné záhlaví má definovanou velikost 12 bajtů. Skládá se z 6 základních položek, které budou popsány níže.
- Sekce dotazů (question) - obsahem tohoto pole jsou DNS záznamy nesoucí požadavky na překlad adres.



Obr. 2.9: Schéma DNS zprávy [25]

- Sekce odpovědí (answer section) - zde se nachází DNS záznamy s výsledky odpovědí na zaslané požadavky.
- Informace o autoritě (authority section) - informace o hlavním (autoritativním) DNS serveru konkrétní sítě.
- Dodatečná sekce (additional section) - nepovinné pole, které bývá vyžito pouze u zpráv typu odpověď. Dodatečné informace mohou být využity k snazšímu překladu.

Hlavička DNS zprávy obsahuje [11, 25]:

- Identifikátor - číselná hodnota, která zajišťuje správné přiřazení odpovědi na příslušný dotaz.
- Příznaky [11, 25]:
  - QR - definuje, zda je zpráva dotazem nebo odpovědí. Pokud je hodnota:
    - 0 – jedná se o zprávu typu dotaz.
    - 1 – jedná se o zprávu typu odpověď.
  - OPCODE - pole, které upřesňuje typ dotazu nebo odpovědi.
  - AA (Authoritative Answer) - identifikuje autoritativní server.
  - TC (Truncation) - značí nutnost rozdělení zpráv na více částí.
  - RD (Recursion Desired) - vyžádání rekurzivní odpovědi od klientské strany.
  - RA (Recursion Available) - rekurzivní odpověď je k dispozici.
  - RCODE (Response code) - informuje o stavu výsledku dotazu. Ukázka možných stavů [11, 25]:
    - 0 – indikuje bezchybný výsledek dotazu.
    - 1 – značí chybu nalezenou ve formátu zprávy.
    - 2 – informuje o chybě, jež je na straně serveru.
    - 3 – oznámení o neexistujícím záznamu,
    - 4 – indikuje typ dotazu, který je nepodporovaný.

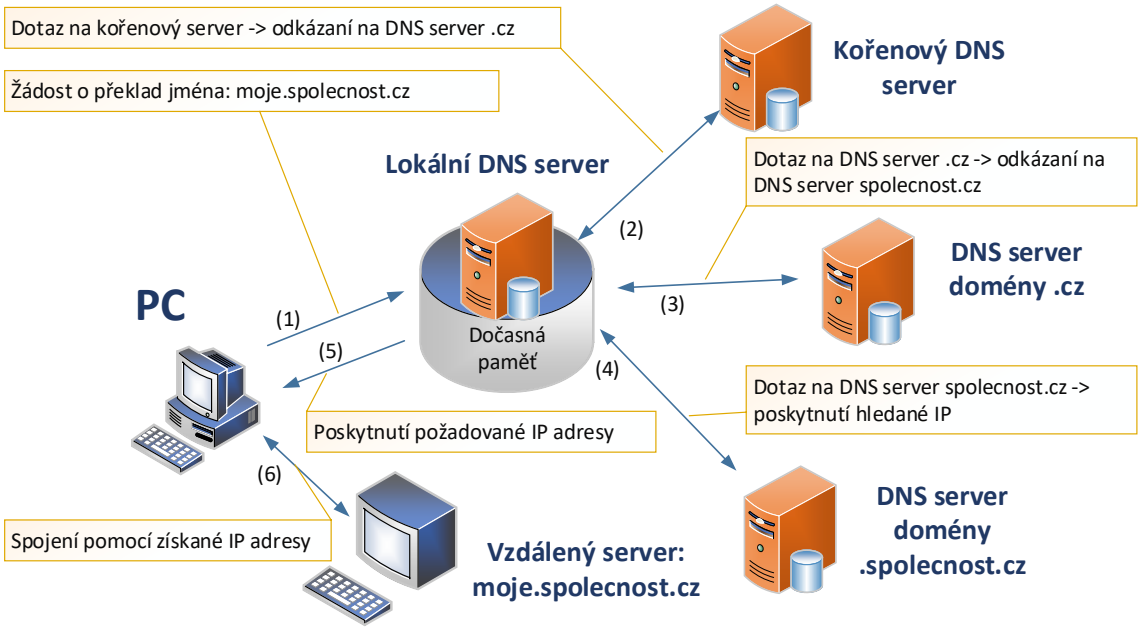
        * 5 – dotaz byl odmítnut.
- Počet záznamů v sekci pro dotazy (QDCOUNT).
- Počet záznamů v sekci pro odpovědi (ANCOUNT).
- Počet záznamu v autoritativní sekci (NSCOUNT).
- Počet záznamů v dodatečné sekci (ARCOUNT).

Princip překladu adres pomocí DNS

Jak již bylo uvedeno, pro překlad IP adres je v případě protokolu DNS využíváno hierarchického postavení DNS serverů [11, 25]. Stejně tak, jsou hierarchicky tvořena i doménová jména, která musejí být stejně jako IP adresy v rámci celosvětové sítě unikátní. Hierarchický systém tvoří stromovou strukturu, přičemž kořenem tohoto stromu je tzv. kořenový (root) server. Kořenový server neodpovídá žádné konkrétní doméně, ale poskytuje odkaz na servery nejvyšší úrovně. Tyto servery, již lze rozeznávat dle unikátního doménového jména. Jako příklad DNS serverů nejvyšší úrovně lze uvést servery spravující domény: .cz, .com nebo .net. Pod tyto servery jsou dále vnořené DNS servery druhé úrovně, na které se příslušné DNS servery odkazují v případě, že se nachází v jejich subdoméně. Doménových úrovní může být více, avšak vždy platí pravidlo, že konkrétní DNS server poskytuje informace pouze o své doméně, nebo odkazuje na DNS servery podřízených domén. Funkci DNS serverů řídí tzv. resolver. Ten na základě žádostí dokáže prohledávat dočasné lokální paměti, kontaktovat jemu nadřízené DNS servery nebo přímo poskytnout příslušné výsledky překladů, jestliže se nacházejí v jeho dočasné paměti [5, 11].

Příklad zjednodušeného DNS překladu se nachází na obrázku 2.10, kde je představeno postupné dotazování pro získání výsledné IP adresy. V daném příkladu stanice PC žádá o překlad doménové jména vzdáleného serveru *moje.spolecnost.cz*. Při odeslání požadavku je nejdříve zkontrolováno, zda není dotazovaná jmenná adresa uložena v lokální paměti stanice PC[5]. Pokud se zde požadované záznamy nenachází, je kontaktován lokální DNS server. Lokální DNS server prohledá svoji dočasnou paměť a jestliže nenalezne hledaný údaj, tak kontaktuje kořenový doménový server. Kořenový doménový server posílá odkaz na DNS nejvyšší úrovně. V našem případě se jedná o doménu *.cz*. Tento DNS server ve své databázi hledaný požadavek nenachází a odkazuje na DNS server nižší úrovně. Tento odkaz vede na DNS server domény *.spolecnost.cz*, která již ve svých záznamech nachází požadovanou IP adresu pro doménové jméno *moje.spolecnost.cz*. Tento záznam je předán na lokální DNS server a následně poskytnut stanici PC. Poskytnutá IP adresa je nyní uložena

[5] V případě OS Windows je tato informace uložena v souboru *\Windows\System32\Drivers\etc\hosts*. Pro linuxové systémy je možné přidat záznamy do souboru */etc/hosts*.



Obr. 2.10: Průběh DNS překladu [5].

v dočasné paměti lokálního DNS serveru po dobu, která je stanovena parametrem TTL v těle příslušné zdrojové věty RR. Stanice PC je nyní schopná kontaktovat vzdálený server pomocí příslušného doménového jména, jelikož nyní zná IP adresu příslušné stanice na základě jejího doménového jména [5, 11].

Výše uvedený příklad popisoval více používanou variantu překladu, kdy je známo doménové jméno a je nutné zjistit jeho IP adresu. Funkce DNS serveru dokáže i opačný proces, kdy je ze známé IP adresy požadováno zjistit doménové jméno. K reverznímu překladu je využíváno fiktivní domény in-addr.arpa [11]. Před tuto fiktivní doménu je navíc nutné zadat IP adresu v opačném pořadí jednotlivých oktetů. Pro žádost o překlad IPv4 adresy 77.75.75.172 je tedy v DNS záznamu (větě RR) uveden doménový název v podobě: 172.75.75.77.in-addr.arpa. Tento záznam je typu 12 (PTR), který značí reverzní překlad doménových jmen [5].

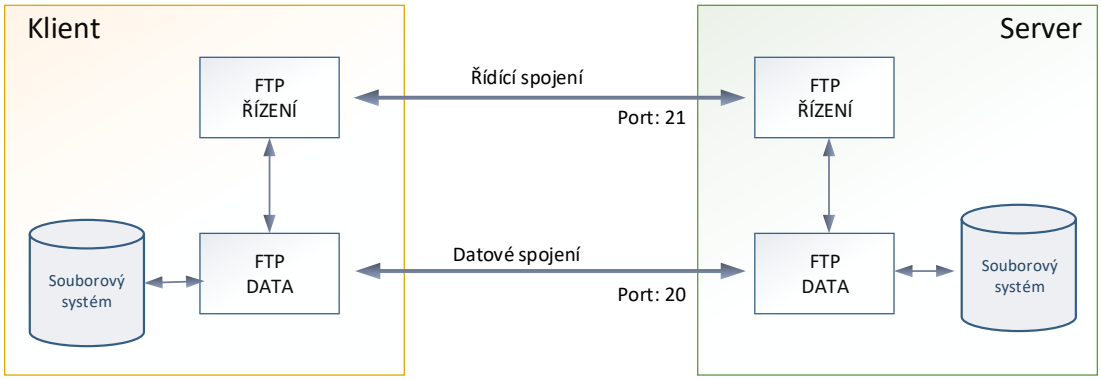
2.3.2 FTP

Protokol FTP (File Transfer Protocol) [23] poskytuje služby umožňující přenos souborů mezi síťovými prvky. Protokol FTP je typickým příkladem služby typu klient-server, který poněkud nestandardně využívá dvou odlišných spojení. Nejdříve je navázáno řídící spojení a až poté je vytvořeno spojení datové. Pro obě spojení se využívá spolehlivého transportního protokolu TCP. Řídící spojení komunikuje implicitně na portu 21 a zahajování tohoto spojení je vždy v kompetenci klienta. Navázané řídící spojení je udržováno po celou dobu komunikace. Datového spojení využívá

implicitně port 20 a bývá navazováno dynamicky v závislosti na aktuální potřebě přenášet konkrétní data, nebo při vykonávání zadaných příkazů. Řídící spojení může být navázáno dvěma různými způsoby [5, 11, 23]:

- Aktivní mód - klient stanoví port, na který server naváže spojení.
- Pasivní mód - server zvolí port, který nabídne klientovi k připojení.

Z důvodů odlišných požadavků na daná spojení využívá FTP protokol právě dvě nezávislá spojení. Schéma spojení FTP protokolu je k vidění na obrázku 2.11. Aby



Obr. 2.11: Schéma komunikace FTP protokolu [5]

mohlo dojít ke stažení dat pomocí protokolu FTP, tak je vyžadováno přihlášení ke vzdálenému FTP serveru. Přihlášení může být provedeno standardně jednoduchou autentifikací pomocí zadání přihlašovacího jména a příslušného hesla. Další možnost je při povolení anonymního režimu, kdy je přihlášení možné jen pomocí zadání veřejně známého přihlašovacího jména a hesla. Přihlášení pod anonymním režimem však velmi často stanovuje omezená práva těmto uživatelům [5, 11].

Protokol FTP nabízí pro práci se soubory i jednoduché uživatelské rozhraní, které je však plně dostačující pro všechny dostupné funkce FTP protokolu. Uživatelům je umožněno procházet adresářové struktury a mají přehled o souborech v daných složkách. K serveru FTP je možné se připojit přes příkazový řádek, pomocí internetových prohlížečů, nebo je možné využít speciálních programů přímo určených pro práci s protokolem FTP. Známým klientem pro FTP připojení je například program FileZilla[6].

Celkově vzato, protokol FTP ve své základní implementaci není nikterak bezpečným protokolem, jelikož například uživatelská hesla lze poměrně snadno zachytit a následně zneužít. V případech, kdy je kladen důraz na zabezpečený přenos, je doporučeno využívat zabezpečené verze protokolu FTP. Příkladem zabezpečené varianty může být protokol FTPS (File Transfer Protocol Secure) [11].

[6]Program FileZilla je volně dostupný software, který je k dispozici na <`https://filezilla-project.org/`>

Příkazy protokolu FTP

Ovládání FTP serveru funguje na principu zasílání řídících příkazů, které jsou vysílány ze strany klienta na FTP server. Jednotlivé příkazy jsou identifikovány posloupností maximálně 4 textových znaků. Vzhledem k velkému množství příkazů, budou v rámci této práce pro ukázku uvedeny jen příklady nejznámějších příkazů [11, 23]:

- Příkazy pro přístup [11, 23]:
  - USER <uživatelské_jméno> - oznamuje přihlášení uživatele.
  - PASS <heslo> - značí zadání hesla.
  - REIN - požaduje novou inicializaci uživatele.
  - QUIT - odhlášení uživatele.
- Příkazy k ovládání funkcí v souborovém systému [11, 23]:
  - CWD <název_složky> - přechod do uvedené složky.
  - CDUP - přechod do nadřazené složky.
  - DELE <název_souboru> - smazání uvedeného souboru.
  - LIST <název_složky> - vypíše obsah aktuálního adresáře.
  - MKD <název_složky> - vytvoření nové složky.
  - PWD - vypíše název aktuálního adresáře.
- Řídící a informativní příkazy [11, 23]:
  - RETR <název_souboru> - stažení souboru ze serveru.
  - STOR <název_souboru> - nahraní souboru na server.
  - PASV - nastaví zahajování datového spojení v pasivním módu.
  - NOOP - kontrola dostupnosti FTP serveru
  - SYST - vyžádá si informace o OS na daném serveru.
  - TYPE <parametr> - dle zadaného parametru definuje typ souboru:
    - <A> - textový dokument.
    - <I> - obrázek.
    - <E> - typ souboru v daném kódovém formátu (EBDCDIC).

Odpovědi protokolu FTP

FTP odpovědi zasílá FTP server klientovi jakožto odezvu na přijaté příkazy. Odpovědi jsou sestaveny z číselného identifikátoru a textové popisu. Číselný identifikátor je složený ze tří číslic <XYZ> umístěných za sebou. První číslice <X> identifikuje 5 stupňů vyhodnocení úspěšnosti daného příkazu [11, 23]:

- 1YZ - předběžné kladné potvrzení, které není finální.
- 2YZ - finální kladné potvrzení.
- 3YZ - průběžné kladné potvrzení, očekávající další příkazy.
- 4YZ - průběžná záporná odpověď, která se však může změnit.
- 5YZ - finální záporná odpověď.

V pořadí druhá číslice <Y> specifikuje 6 možností, čeho se předchozí vyhodnocení úspěšnosti týká [23]:

- X0Z - syntaxe.
- X1Z - informace.
- X2Z - spojení.
- X3Z - autentizace či uživatelského účtu.
- X4Z - nespecifikováno.
- X5Z - souborového systému.

Poslední číslice <Z> poskytuje další doplňující informace [23].

2.3.3 TFTP

TFTP (Trivial File Transfer Protocol) [24] je velice zjednodušená verze předchozího protokolu FTP, jež se používá především pro přenos malých souborů, které dle základní specifikace nesmí být větší než 32 MB. Na rozdíl od FTP protokolu, se v případě TFTP protokolu používá nespojovaný a nespolehlivý přenos pomocí protokolu UDP. Standardně je protokol TFTP možné identifikovat pod portem 69. Přenášená data jsou rozdělena do bloků o velikosti 512 bajtů [7], která jsou postupně číslována a přenášena. Číslování jednotlivých bloků typicky začíná hodnotou 1. Po odeslání jednotlivého bloku musí dojít k potvrzení jeho doručení. Bez tohoto potvrzení tak není možné odeslat následující blok. Pokud nedojde k potvrzení daného bloku před vypršením daného limitu, musí dojít k novému odeslání nedoručeného bloku dat. Transfer dat je u konce, jakmile je potvrzený poslední blok dat, který se vyznačuje tím, že má menší velikost než je stanovený maximální limit velikost bloku dat. Kontinuální potvrzování jednotlivých bloků dat má za následek také výslednou nižší přenosovou rychlost v porovnáním s protokolem FTP [11, 24]. Dle výše zmíněných vlastností protokolu TFTP je patrné, že v rámci globálního použití běžnými uživateli, není použití protokolu TFTP perspektivní. Proto se protokol TFTP používá především v lokálních sítích. Jeho vlastnosti najdou uplatnění například při kopírování konfiguračních dat síťových prvků. Typickým příkladem je také využití u bezdiskových stanic, kdy jsou do operačních systému pomocí protokolu TFTP vkládána data, která slouží ke spuštění daných systémů [11].

Zjednodušení protokolu se projevilo také na možnostech připojení k daným TFTP serverům. TFTP server nevyužívá žádného uživatelského přihlašování, stejně tak neumožňuje žádné uživatelské rozhraní umožňující například procházení adresářových struktur. Stažení dat probíhá pomocí zadání příkazu v příkazové řádce, kde musí

[7] Dle nejnovější specifikace TFTP protokolu (RFC 7440) z roku 2015 je možné navýšit maximální velikost bloků dat. Hraniční velikost by však neměla být vetší než je stanovená hodnota MTU, aby nedocházelo k fragmentaci na síťové vrstvě.

být přesně specifikována cesta s názvem požadovaného souboru a příslušné parametry [11, 24].

Typy zpráv protokolu TFTP

Protokol TFTP ke svému fungování využívá 5 základních typů zpráv [11, 24]:

- Zpráva RRQ - navazuje spojení pro čtení souboru ze serveru. V těle této zprávy se nachází identifikátor typu zprávy (OpCode), který má hodnotu 1. Dále se zde nachází název požadovaného souboru a taky položka MODE, která určuje zda je tento soubor binární, nebo se jedná o textový dokument.
- Zpráva WRQ - navazuje spojení pro nahrávání souborů na server. Tělo zprávy je totožné s předchozí zprávou. Liší se pouze identifikátor, který má nyní hodnotu 2.
- Zpráva DATA - nese samotná data. V těle zprávy se nachází položka nesoucí číslo daného bloku a položka, která nese samotná data. Nechybí také identifikátor zprávy, který nese hodnotu 3.
- Zpráva ACK - je kladná potvrzovací zpráva, která s sebou nese číslo potvrzeného bloku. Pole OpCode má hodnotu 4.
- Zpráva ERROR - chybová zpráva, která informuje o případné nemožnosti navázání spojení (odpověď na zprávy RRQ nebo WRQ), nebo problému při přenosu datových zpráv. Tělo zprávy s sebou nese identifikátor (OpCode) s nastavenou hodnotou 5, a také číselný identifikátor upřesňující typ chyby s upřesňujícím textovým popisem. Chyby mohou být typu [11, 24]:
  - 0 - nespecifikovaná chyba,
  - 1 - požadovaný soubor nebyl nalezen,
  - 2 - narušení přístupu,
  - 3 - není volné místo na disku,
  - 4 - nepovolená operace,
  - 5 - neznámé číslo portu,
  - 6 - daný souboru již existuje,
  - 7 - přihlášení uživatele, který je neplatný.

3 Výběr a popis testovacího prostředí

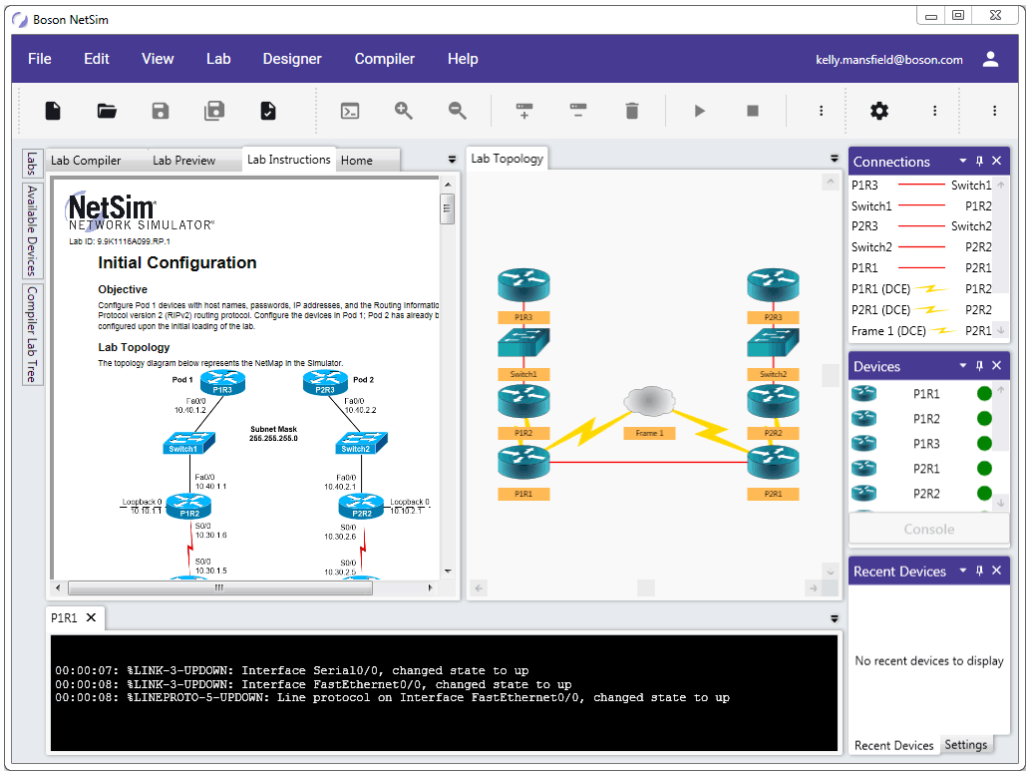
V následující kapitole budou představeny testovací prostředí, které by svými vlastnostmi byly vhodné pro realizaci laboratorních úloh. Následně bude popsána příprava a instalace vybraného laboratorního prostředí.

3.1 Výběr testovacího prostředí

Důležitými kritérii pro volbu testovacího prostředí bude možnost přehledně demonstrovat základní vlastnosti síťových protokolů, jednoduchá konfigurace prvků, dostupné materiály a dokumentace. Neméně významným kritériem je také bezesporu kvalita a dostupnost vývojářské podpory. Na základě získaných poznatků bude zvoleno vhodné testovací prostředí.

3.1.1 Boson Network Simulator

Síťový simulátor od společnosti Boson je zaměřený především na simulaci síťových topologiích testující převážně zařízení Cisco. Simulační nástroj Boson bývá využíván k certifikaci a školení v oblasti IT. Náhled grafického prostředí zmíněného programu se nachází na obrázku 3.1.



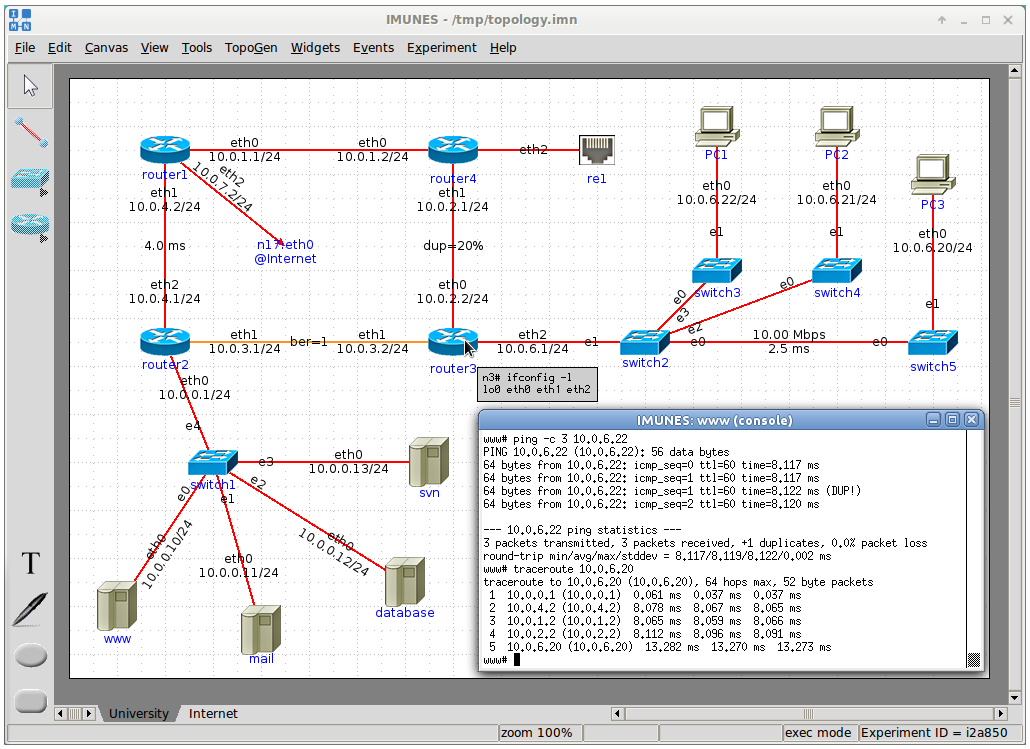
Obr. 3.1: Ukázka prostředí Boson Network Simulator [26].

V našem případě stojí za zmínku především schopnost jednoduše vytvářet laboratorní úlohy, věnující se síťové problematice. Do laboratorních úloh lze snadno a přehledně implementovat jednotlivé postupy a v propracovaném grafickém prostředí tak lze využít možností jako je aktivní historie úprav, přidávání komentářů k jednotlivým postupům a interaktivní ovládání [26].

Velkou nevýhodou je fakt, že všechny verze simulátoru Boson Network jsou pouze komerční. Cena základní licence pro jednoho uživatele je stanovena na 99 dolarů [26].

3.1.2 IMUNES

Univerzální nástroj IMUNES, jehož zkratka je složená z anglických slov: Integrated Multiprotocol Network Emulator Simulator, slouží k simulaci síťových topologií. Emulátor IMUNES umožňuje generovat síťový provoz IP sítí v reálném čase při rychlostech dosahujících až 1 Gb/s, přičemž dokáže simulovat až stovky síťových uzlů. Technologie virtualizace síťového zásobníků použitého v tomto nástroji byla vyvinuta na univerzitě v Záhřebě. K dispozici je rozsáhlá dokumentace podrobně popisující všechny funkce softwaru a obsahuje také návody na realizaci simulace jednoduchých sítí [27]. Náhled grafického prostředí je na obrázku 3.2.



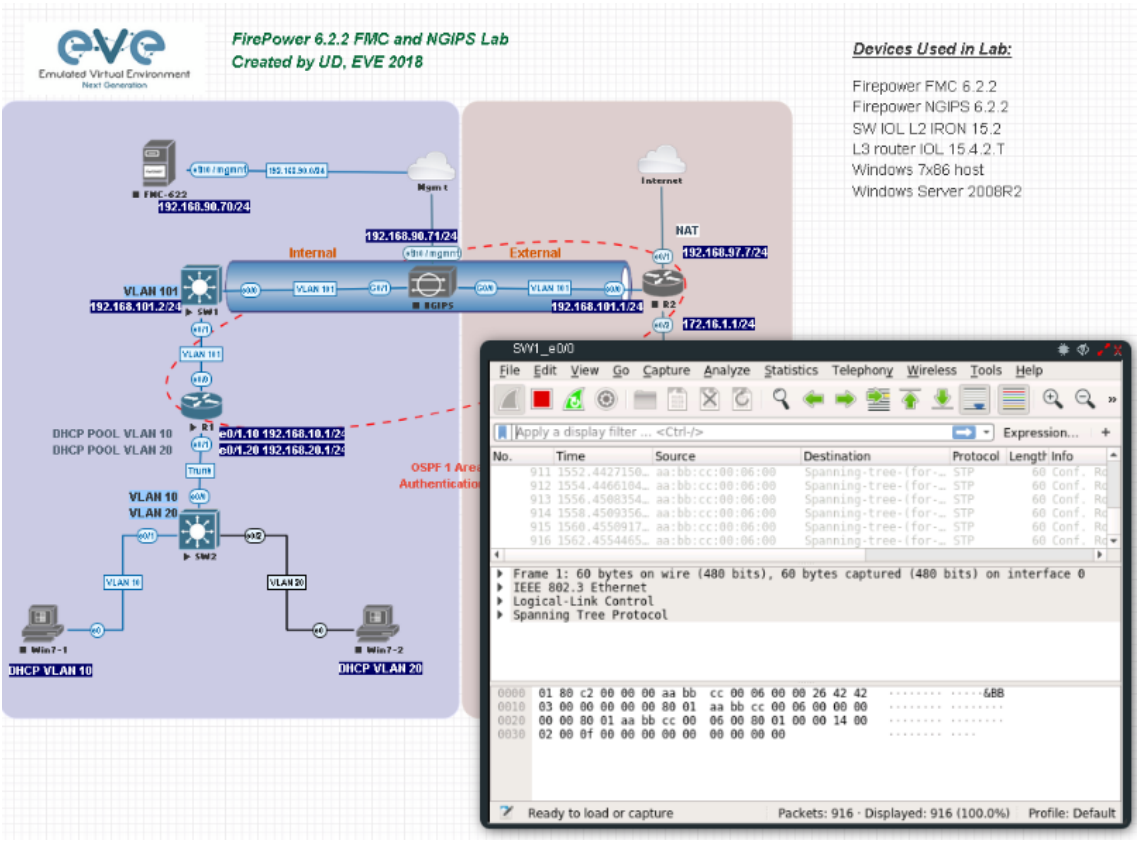
Obr. 3.2: Ukázka prostředí IMUNES [27].

IMUNES je dostupný jako FreeBSD systém spustitelný přes nástroj VirtualBox, nebo jej lze doinstalovat do systémů Ubuntu. Simulovaný síťový provoz lze

mezi dvěma komunikujícími entitami analyzovat zabudovaným nástrojem Wireshark. IMUNES je nabízený jako open-source [27].

3.1.3 EVE-NG

EVE-NG je síťový emulátor, který podporuje virtuální verze komerčně prodávaných routerů značky Cisco, Juniper, CheckPoint apod. Software umožňuje na základě generování reálného provozu známých síťových protokolů a pomocí zakomponovaných nástrojů otestovat navržená schémata řešení. Emulátor EVE-NG vyniká vlastnostmi jakou je vlastní podpora jádra pro L2 protokoly, možnost importu a exportu síťové konfigurace, nebo paměťovou optimalizací [28].



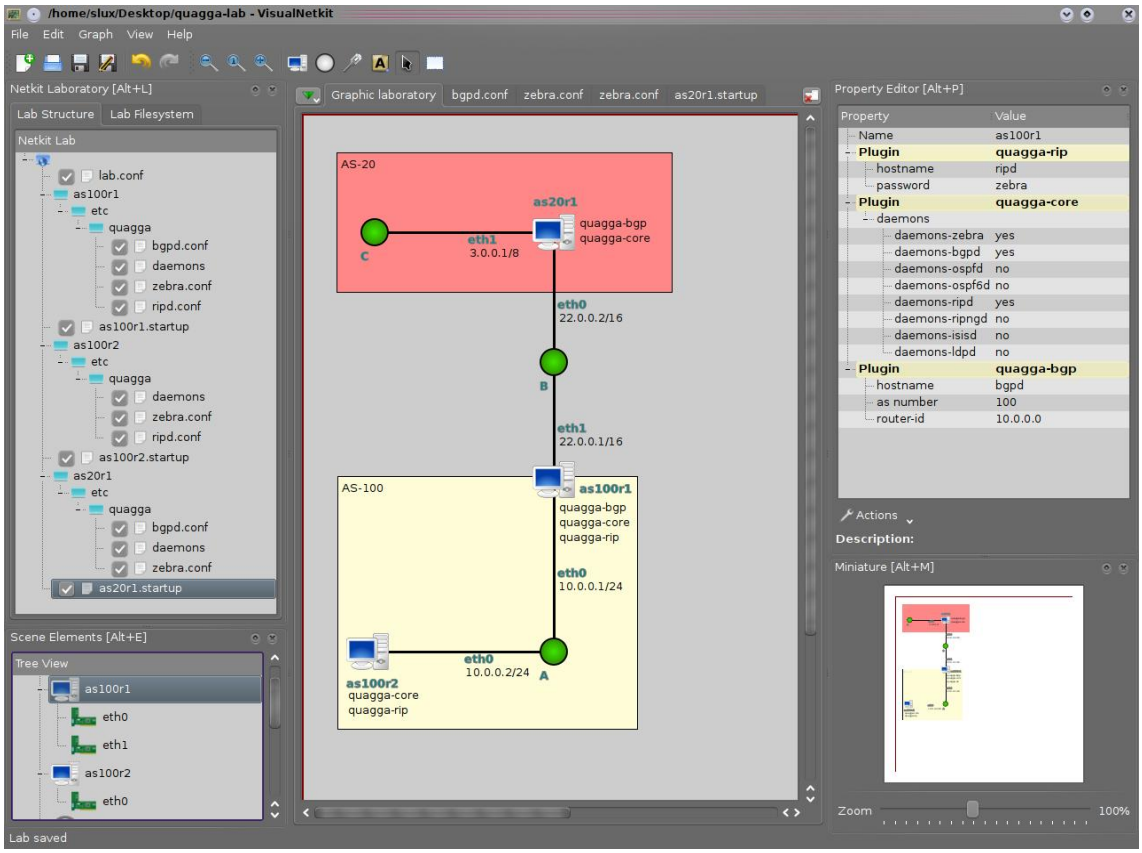
Obr. 3.3: Ukázka prostředí EVE-NG [28].

Emulátor je možné zakoupit a nainstalovat jako virtuální stroj ze souboru ISO běžící jako součást operačního systému Ubuntu. EVE-NG je k dispozici v placené PRO-Edition verzi, nebo ve vydání pro komunitu, která je zdarma. Komunitní verze však oproti placené verzi neobsahuje všechny diagnostické nástroje a práce se softwarem je v neplacené verzi značně složitější [28].

3.1.4 Netkit

Otevřený síťový simulátor Netkit byl vytvořen pro studijní potřeby, jako nástroj sloužící k výuce. Jedná se o software s otevřeným zdrojovým kódem založený na linuxovém jádře. K dispozici je nabízeno množství předkonfigurovaných laboratorních scénářů i s přednáškovými snímky, popisující konkrétní scénář a technologii použitou v daném scénáři. Knihovna s předkonfigurovanými laboratorními scénáři je dostupná na dostupné webové stránce Netkitu a je jednou z klíčových výhod tohoto simulačního nástroje [29].

Podobně jako jiné síťové simulační nástroje poskytuje Netkit sadu příkazů k vytvoření virtuálních strojů, které jsou vzájemně propojeny virtuální sítí, zahrnující například směrovací technologie (RIP, OSPF), simulaci několika typů serverů (FTP, HTTP, SMTP), firewally, diagnostické nástroje jako ping, traceroute, tcpdump, apod. Program Netkit je možné používat a ovládat pomocí příkazové řádky nebo je možné doinstalovat plugin pro grafické prostředí s názvem VisualNetkit (viz obrázek 3.4 [29].



Obr. 3.4: Ukázka prostředí Netkit [29].

3.1.5 NS-3

Platforma NS-3 patří do skupiny síťových simulátorů určených především pro vzdělávací a výzkumné účely. Jedná se o otevřené simulační prostředí s velmi dobrou podporou vývojářů a propracovanou dokumentací [30].

Simulační jádro NS-3 umožňuje výzkum a vývoj simulačních modelů, které jsou dostatečně reálné, aby umožnily použití NS-3 jako emulátoru síťových protokolů v reálném čase a případné propojení se skutečnou infrastrukturou. Tvorba simulací probíhá pomocí programovacího jazyka C++ a Python. Simulovat je možné nepřeberné množství transportních a komunikačních protokolů jako je například protokol TCP, UDP, ARP a spoustu dalších. Veškerý síťový provoz pak lze přehledně kontrolovat pomocí směrovacích tabulek, ale také jej lze zachytávat a později analyzovat pomocí zakomponovaného nástroje Wireshark [30].

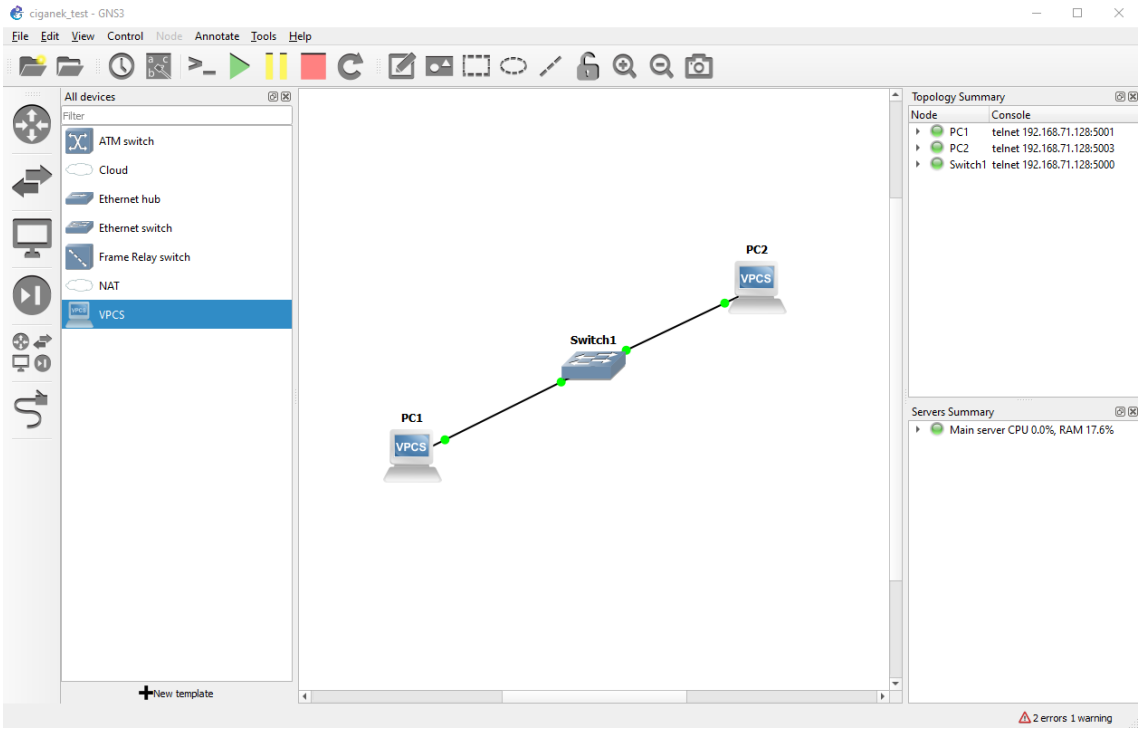
NS-3 je open-source nástroj, jež je licencovaný na základě licence GNU GPLv2. NS-3 je kompatibilním s operačními systémy Linux. Jak již bylo zmíněno, velkou výhodou je velké množství funkcí, kvalitní dokumentace a podpora vývojářů. Za nedostatky je možné považovat chybějící grafické uživatelské prostředí. Ke konfiguraci simulací je tedy nutná dobrá znalost programovacího jazyka C++ nebo Python [30].

3.1.6 GNS-3

GNS3 je open-source simulační nástroj s propracovaným grafickým prostředím a podobně jako u NS-3 s velmi obsáhlou dokumentací a podporou vývojářů [31].

GNS-3 je často využíván nejen pro studijní účely, ale slouží i pro přípravu k Cisco certifikacím CCNP či CCNA. Využití je možné také pro testování síťového provozu, jelikož je možné simulované prostředí jednoduše připojit do reálné sítě. Do vytvořených simulací lze také zakomponovat jakékoliv virtuální zařízení na platformách VMware nebo VirtualBox. Nástroj podporuje celou řadu síťových prvků různých značek a typů. Za zmínku stojí například známé směrovače značek Cisco, Juniper a Mikrotik a spousta dalších typů síťových zařízení jako jsou přepínače, firewally a koncové uživatelské stanice [31].

Po registraci na webovém portálu je možné zdarma stáhnout GNS-3 ve verzích pro platformy Windows, Mac a Linux. V době psaní této práce je aktuální verze označena GNS-3 v2.2.0. Pro optimální výkon vývojáři doporučují stažení doplňkové virtualizace GNS-3 VM běžící na OS Ubuntu 18.04.3. Ukázka grafického prostředí GNS3 je k vidění na obrázku 3.5[31].



Obr. 3.5: Ukázka grafického prostředí GNS-3.

3.1.7 Shrnutí výběru laboratorního prostředí

Na základě získaných informací bylo pro tvorbu laboratorních úloh vybráno simulační prostředí GNS3. Jak již bylo zmíněno v kapitole 3.1.6 jedná se o open-source software s propracovaných grafickým prostředím, který dokáže simulovat často využívaná síťová zařízení. GNS3 splňuje všechny požadované vlastnosti a proto bude GNS3 zvoleno pro realizaci laboratorních úloh.

3.2 Příprava testovacího prostředí

Jak bylo uvedeno v kapitole 3.1.6, prostředí GNS-3 je možné instalovat na 3 platformy a to na OS Windows, Linux a MAC. Pro vyhnutí se problematiky případného licencování operačního systému, bylo vybráno prostředí Linux a to konkrétně operační systém Ubuntu 18.04.3 LTS. Aby bylo možné vypracované laboratorní úlohy bez problému přenášet na různá pracoviště a laboratorní učebny, bude výhodné celé simulační prostředí GNS3 virtualizovat. K virtualizaci bude použit nástroj VMware® Workstation 15 Pro. Tento krok zaručí zmiňovanou výhodu přenositelnosti, ale také možnost zachycení konkrétní konfigurace operačního systému, následnou okamžitou obnovu a spoustu dalších výhod.

3.2.1 Instalace GNS3

Po úspěšné instalaci operačního systému Ubuntu ve virtuálním prostředí VMware je možné instalovat simulační prostředí GNS3. Po spuštění příkazové řádky pomocí aplikace Terminal je možné postupně zadávat následující příkazy [32]:

- Přidání instalačního baličku GNS3 a jeho následná instalace.

```
sudo add-apt-repository ppa:gns3/ppa
sudo apt update
sudo apt install gns3-gui gns3-server
```

- Umožnění instalaci aplikací běžících pro 32 bitové operační systémy.

```
sudo dpkg --add-architecture i386
```

- Přidání balíčku IOU umožňující především simulaci zařízení značky Cisco.

```
sudo apt install gns3-iou
```

- Smazaní všech předchozích balíčků Docker, import certifikátů a povolení přístupu ke stahování dalších komponentů GNS3.

```
sudo apt remove docker docker-engine docker.io
sudo apt-get install apt-transport-https ca-
   certificates curl \ software-properties-common
```

- Přidání balíčku Docker umožnující spouštět síťové prvky, které fungují na této platformě.

```
sudo apt update
sudo apt install docker-ce
```

- Přidání vytvořeného účtu student do složek ubrifge, libvirt, kvm a wireshark.

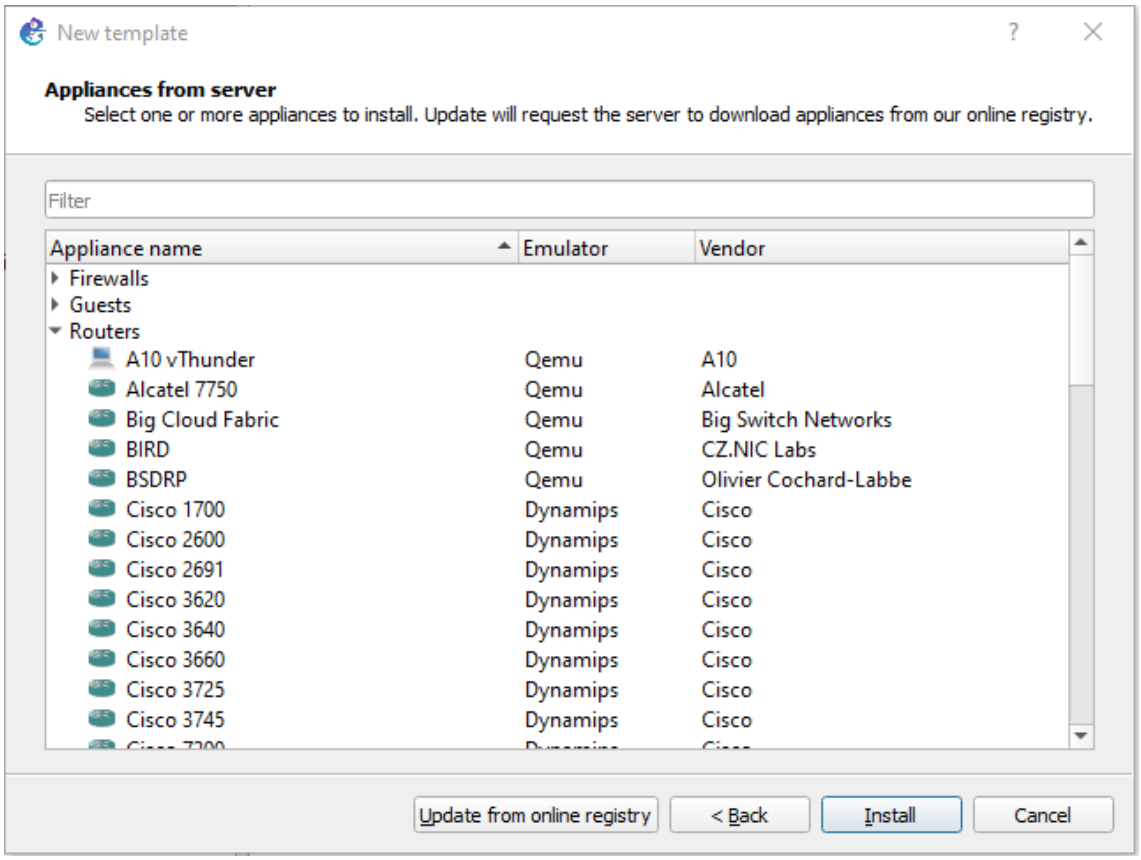
```
sudo usermod -aG ubrigge student
sudo usermod -aG libvirt student
sudo usermod -aG kvm student
sudo usermod -aG wireshark student
```

Po výše uvedeném postupu je simulační prostředí připraveno ke spuštění. Vzhled GNS3 po kompletní instalaci reprezentuje obrázek 3.5. V základním nastavení se zde nachází 7 předinstalovaných síťových prvků:

- ATM switch,
- Cloud,
- Ethernet hub,
- Ethernet switch,
- Frame Relay switch,
- NAT,

- VPCS.

Dle potřeby je zde možné požadované síťové prvky importovat pomocí souborů ve formátu .qcow2. Import lze provést přímým vložením zachyceného souboru, nebo je možné instalační soubory síťových prvků stáhnout a instalovat přímo ze serverů GNS3. Ukázka výběru simulovaných zařízeních se nachází na obrázku 3.6. Tímto poměrně jednoduchým způsobem lze simulovaná zařízení přidávat. Po přidání zařízení je možné konkrétní zařízení přetáhnout do pracovního pole programu a začít s ním pracovat.



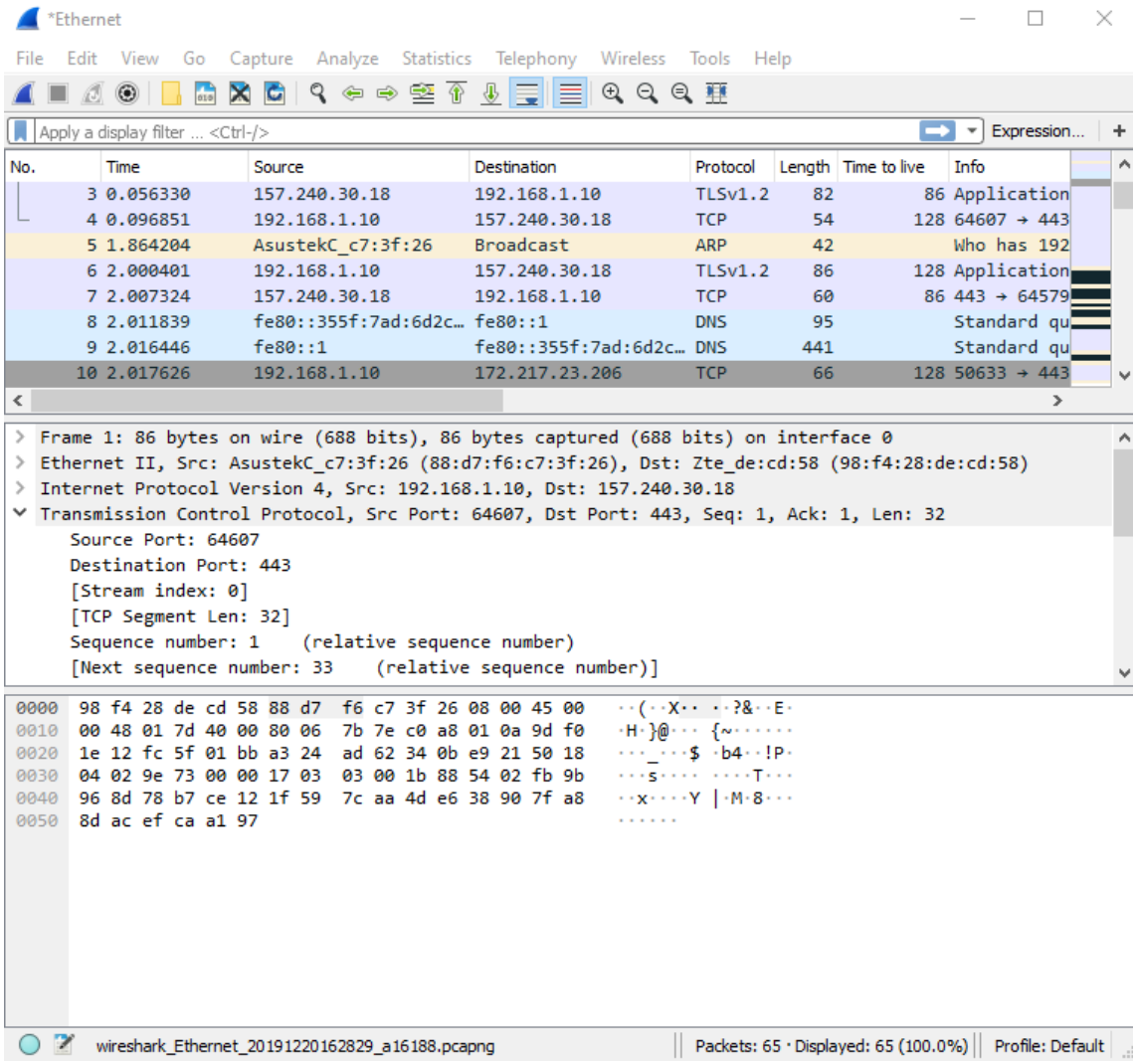
Obr. 3.6: Výběr simulovaných prvků k instalaci v GNS3

3.3 Popis použitých síťových nástrojů

3.3.1 Wireshark

V dnešní době je program Wireshark [33] jeden z nejpoužívanějších softwarů, sloužící k zachycení, monitoringu a analýze síťové komunikace. Wireshark umožňuje na zvoleném rozhraní zachytit pakety probíhající síťové komunikace a tyto pakety pak do detailů analyzovat. Wireshark je open-source software s velkou podporou

komunity a vývojářů. K dispozici je propracované grafické rozhraní s celou řadou funkcí jako například filtrace a vyhledávání paketů, import do textových souborů, či grafické vykreslení síťové komunikace dle mnoha kriterií. V případě, že není dostupné grafické uživatelské rozhraní je možnost využít verzi Wiresharku pojmenovanou TShark, která umožňuje zachytávat pakety na strojích nebo terminálech pouze za pomoci příkazové řádky [34]. Ukázka prostřední programu Wireshark je k vidění na obrázku 3.7.



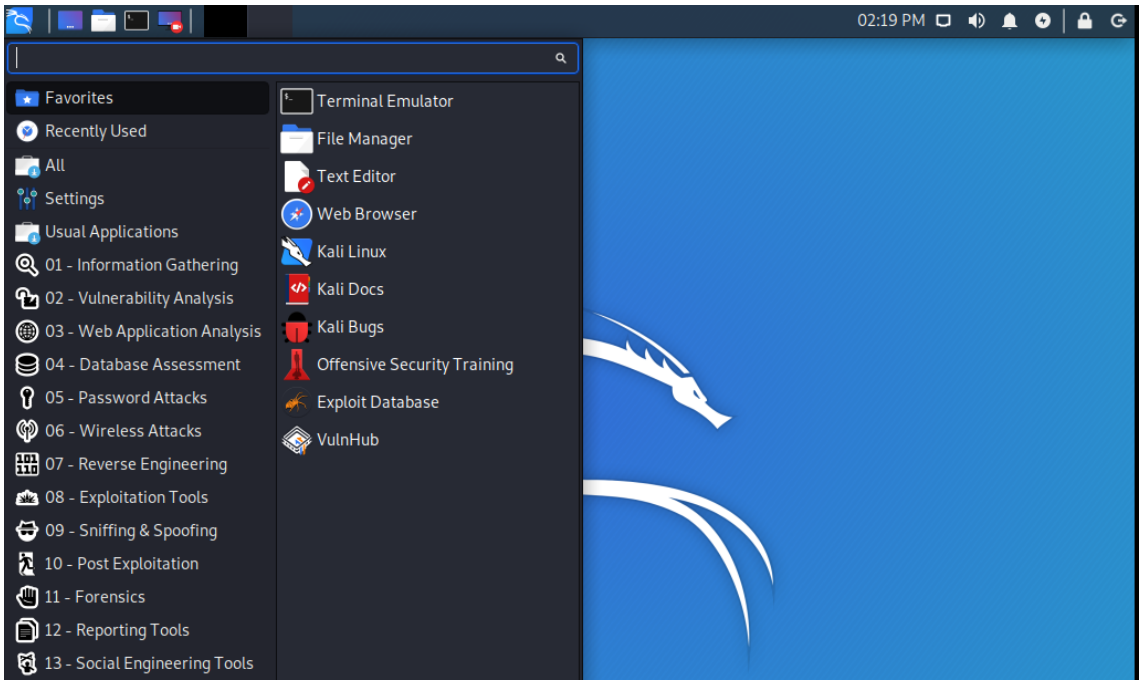
Obr. 3.7: Ukázka prostředí programu Wireshark.

3.3.2 Kali Linux

Kali Linux [39] je volně dostupná linuxová distribuce, která obsahuje spoustu nástrojů sloužící k tvorbě bezpečnostních analýz sítí a také k tzv. penetračnímu testování. Mezi nejznámější nástroje, které jsou součástí Kali Linux patří zejména [39]:

- Nmap,
- Metasploit Framework,
- Hydra,
- Aircrack–ng.

Za vývojem distribuce Kali Linux stojí komunita IT specialistů, kteří na svém webu [1] poskytují veškeré informace včetně dokumentace produktu, aktivního fóra i tutoriálů zaměřující se na práci s Kali Linux [39]. V rámci laboratorní úlohy bude použita nejnovější verze z března roku 2020 označená jako Kali Linux 2020.1a. Ukázka prostředí virtualizace Kali Linux, jež bude využita v 3. laboratorní úloze, je na obrázku 3.8.



Obr. 3.8: Ukázka prostředí Kali Linux.

[1] https://www.kali.org

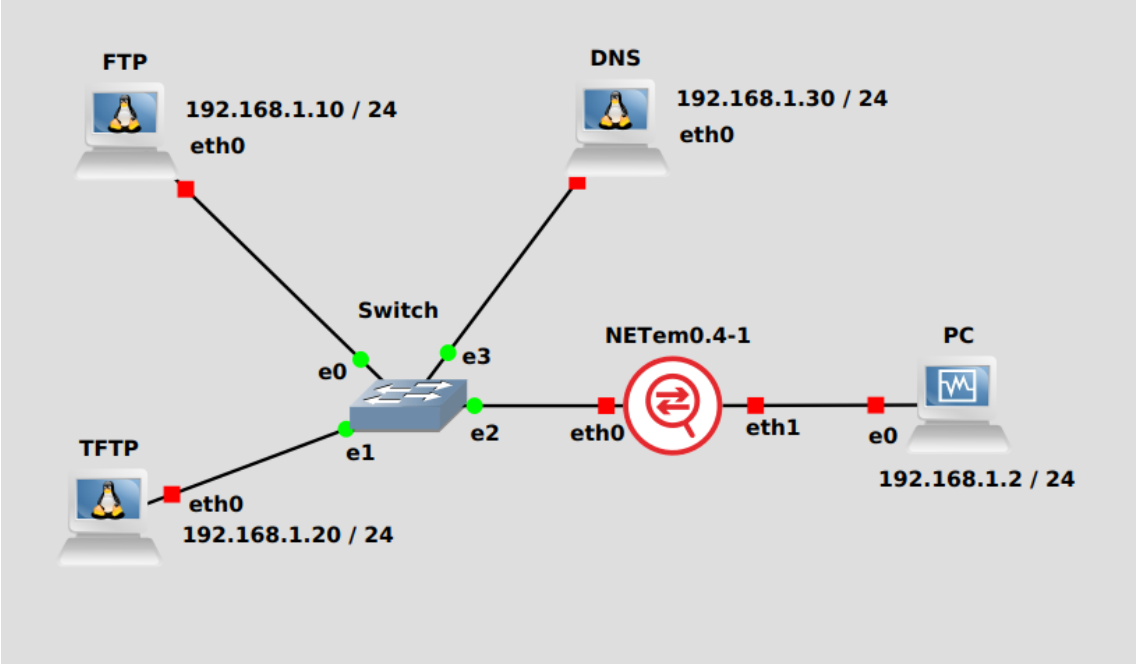
4 Návrh laboratorních úloh

Tato kapitola se zabývá podrobným popisem návrhů laboratorních úloh, které se věnují základním vlastnostem síťových protokolů. Uvedené úlohy jsou navrženy a realizovány ve zvoleném simulačním prostředí GNS3. K analýze síťové komunikace je ve všech úlohách využito programu Wireshark.

4.1 Laboratorní úloha: Porovnání transportních protokolů TCP a UDP

4.1.1 Popis laboratorní úlohy

Hlavním úkolem laboratorní úlohy je v simulačním prostředí GNS3 demonstrovat základní rozdíly transportního protokolu TCP a UDP. K simulaci transportních protokolů využijeme komunikaci aplikační protokolů FTP a TFTP. Studenti budou mít v první řadě za úkol seznámit se se simulačním prostředím GNS3 a dle návodu vytvořit funkční zapojení, umožnující analýzu síťového provozu pomocí nástroje Wireshark. Analýza je zaměřena především na vlastnosti transportních protokolů. Schéma zapojení jednoduché sítě v prostředí GNS3 se nachází na obrázku 4.1.



Obr. 4.1: Schéma zapojení pro demonstraci rozdílů TCP a UDP.

Po správné konfiguraci je možné na virtuálním PC provést stažení testovacího souboru ze serverů FTP a TFTP. Při stahování souboru pak bude jednotlivá komunikace analyzována. Nejdříve bude zkoumána probíhající TCP komunikace ze serveru FTP na klientskou stanici. Následně bude stejný soubor stahován ze serveru TFTP, při kterém bude analyzován transportní protokol UDP.

Aby bylo možné plně demonstrovat rozdíly mezi transportními protokoly, je využito možnosti regulace kvalitativních vlastností přenosového spojení mezi komunikujícím PC a servery. K tomu nám poslouží nástroj NETem, který dokáže nastavit vlastnosti přenosového kanálu, kterými jsou maximální šířka pásma, ztrátovost a zpoždění. Po zvládnutí všech bodů laboratorní úlohy budou mít studenti za úkol ze získaných poznatků definovat hlavní rozdíly mezi protokoly TCP a UDP.

4.1.2 Konfigurace použitých síťových prvků

FTP server - Toolbox

Pro simulaci FTP serveru bude použit nástroj Toolbox [35], který lze volně stáhnout a importovat ze serveru GNS3. Jedná se o zjednodušenou verzi operačního systému Ubuntu, kde jsou nainstalované pouze základní komponenty, které slouží k vytváření jednoduchých serverů (WWW, FTP, TFTP, DHCP, Syslog a SMNP) v prostředí GNS3. K ovládání a konfiguraci Toolboxu se používá připojení Telnet.

Pro základní konfiguraci, která bude v rámci laboratorní úlohy dostačující, je nutné po připojení ke konzoli FTP serveru upravit soubor **/etc/vsftpd.conf** do následující podoby:

```
listen=YES #umožní samostatné spuštění služby vsftpd
anonymous_enable=NO #pro přístup k souborům je vyžadováno
    přihlášení
local_enable=YES #kontrola práv pro přístup na lokálním
   serveru
chroot_local_user=NO   #po přihlášení je konkrétním uži-
   vatelům umožněn přístup do soukromých složek
write_enable=YES #povolení ovládání pomocí FTP příkazů
local_umask=022 #nastavení plných práv lokálním uživatelů
    pro vytváření souborů
pam_service_name=vsftpd #název služby FTP
```

IP adresu FTP serveru je možné nastavit pomocí příkazu:

```
ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
```

Po změně konfigurační souboru je nutné službu FTP serveru restartovat. Učinit se tak může následujícími příkazy:

```
/etc/init.d/vsftpd stop
/etc/init.d/vsftpd start
/etc/init.d/vsftpd restart
```

Soubory, které se budou používat ke stažení na klientskou stanici, je nutné nejdříve nahrát do složky **/root** vytvořené instance Toolbox. Do adresářové složky jednotlivých prvků je přístup možný pomocí kliknutí pravého tlačítka myši na ikonu daného zařízení a zvolením položky Show in file manager.

TFTP server - Toolbox

K simulaci TFTP serveru bude použita, stejně jako v případě FTP serveru, instance Toolbox [35]. Nyní se však využije funkce tftpd-hpa. Po přetažení instance do pracovní plochy GNS3 se je možné připojit přes rozhraní Telnet. Po připojení ke konzoli síťového prvku je možné začít s konfigurací stanice. Požadovanou IP adresu je možné nastavit pomocí příkazu:

```
ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
```

Konfigurační soubor pro TFTP server **/etc/default/tftpd-hpa** se nastaví následovně:

```
TFTP_USERNAME="tftp"  #název služby TFTP
TFTP_DIRECTORY="/tftpboot" #složka, obsahující stahované
   soubory
TFTP_ADDRESS="192.168.1.20:69" #nastavení IP adresy a
   portu transportní služby
TFTP_OPTIONS="--secure --create"    #nastavení umožnující
    přístup pouze ke konkrétnímu souboru, povolení nahrá-
   vání nových souborů
```

Podobně jako u FTPT serveru je při změně konfigurace nutné restartovat služby TFTP serveru:

```
service tftpd-hpa stop
service tftpd-hpa start
service tftpd-hpa restart
```

Soubory, které se budou stahovat na klientskou stanici ze serveru TFTP je možné nahrát stejným způsobem jako u předešlého FTP serveru s tím rozdílem, že cílová složka je **/tftpboot**.

DNS server

Aby si studenti mohli otestovat i fungování protokolu DNS, je v testovacím prostředí vytvořen lokální DNS server, který bude umět překládat IP adresy FTP a TFTP serveru na zvolená doménová jména. Prostředí GNS3 nabízí možnost použití instance s názvem DNS [36], která má na základním linuxovém jádře OS Ubuntu předinstalovanou funkci dnsmasq zajišťující překlad IP adres. Po stažení a importu je nástroj DNS plně k dispozici. Po přetažení a spuštění zařízení je možné se připojit na konzoli pomocí připojení Telnet a začít s konfigurací. IP adresu je možné nastavit úpravou souboru **/etc/network/interfaces** do následující podoby:

```
auto eth0  #zpřístupnění fyzického portu eth0
iface eth0 inet static #na rozhraní eth0 povolí manuální
   nastavení pro adresy ipv4
address 192.168.1.30   #zadání ipv4 adresy
netmask 255.255.255.0   #zadání masky podsítě
up echo nameserver 192.168.1.30 > /etc/resolv.conf
#zadání IP adresy DNS serveru a zapsání do konf. souboru
```

Výše uvedenou konfigurací bude nastaveno síťové rozhraní DNS serveru. Poslední příkaz zaručí, že ostatní zařízení v síti mohou o překlad IP adres požádat na zadané IP adrese.

Nyní můžeme přistoupit k nastavení seznamu IP adres, kterým se přiřadí volitelná doménová jména. Seznam adres se nachází v souboru **/etc/hosts**, do kterého se připíší požadované IP adresy a zvolená doménová jména ve formátu [IP] [doménové jméno]. V případě vytvořené laboratorní úlohy bude soubor vypadat následovně:

```
192.168.1.10 muj_ftp #překlad adresy FTP serveru
192.168.1.20 muj_tftp #překlad adresy TFTP serveru
```

Po úpravě souboru je nutné jej uložit a restartovat službu DNS serveru dnsmasq: Učinit tak je možné příkazy:

```
/etc/init.d/dnsmasq stop
/etc/init.d/dnsmasq start
/etc/init.d/dnsmasq restart
```

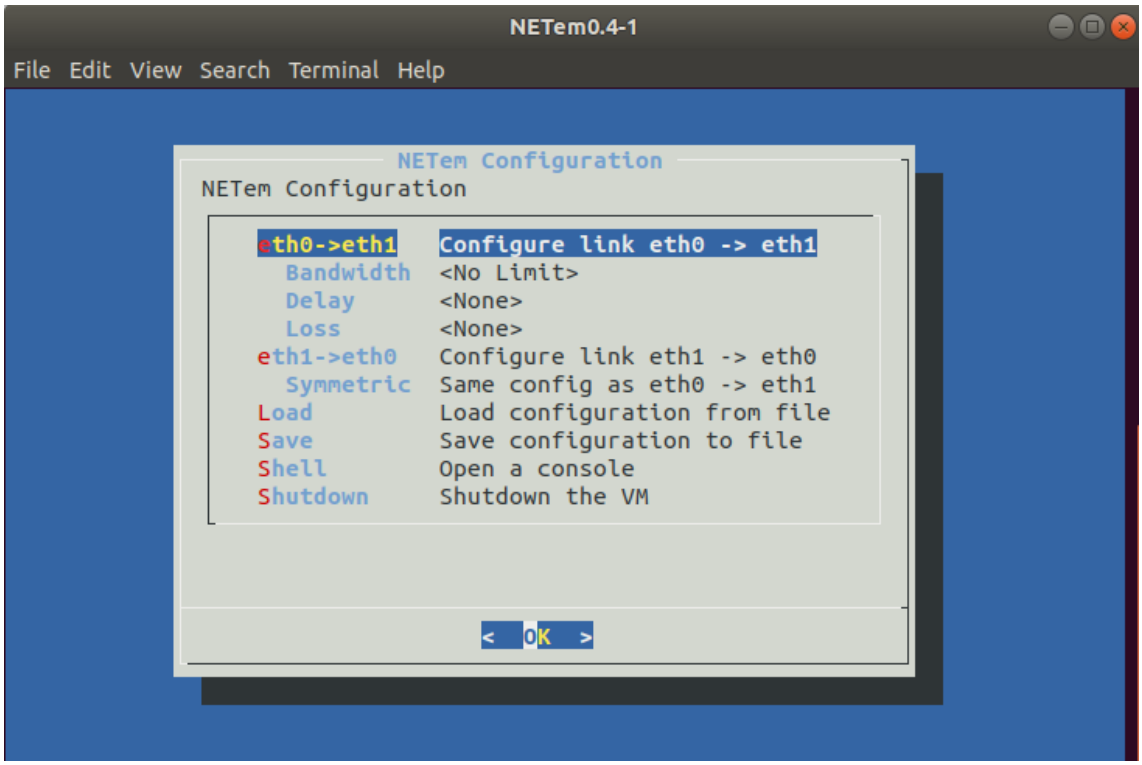
Přepínač

K přehlednému propojení všech síťových prvků v navrženém schématu bude využito přepínače. Jak již bylo uvedeno v kapitole 3.2.1, v prostředí GNS3 je možné využít připraveného prvku ethernetového přepínače, který je jedním ze základních prvků

dostupných ihned po instalaci GNS3. Instanci přepínače není nutné nikterak konfigurovat. Po přetažení na pracovní plochu je možné připojit jakýkoliv prvek pouhým kliknutím (při spuštěném režimu **Add a link**) a vybráním volného fyzického portu.

Regulační prvek pro parametry síťové linky - NETem

Aby bylo možné ověřit vlastnosti transportních protokolů při změnách vlastností přenosového vedení, bude nutné nějakým způsobem regulovat vlastnosti přenosu navrženého testovacího schématu. Pro tento požadavek je v simulačním prostředí GNS3 možné použít instanci NETem 0.4 [37], která je stejně jako Toolbox volně dostupná na serverech GNS3. Po stažení a importu se instance jednoduše vloží mezi dva síťové prvky a propojí se. V našem případě bude NETem umístěn mezi klientskou stanicí a přepínačem. NETem je plně transparentní a přemosťuje síťový provoz z jednoho portu na druhý. Po připojení ke konzoli se v uživatelsky přívětivém prostředí mohou měnit vlastnosti parametrů linky (viz obrázek 4.2). Konkrétně lze měnit parametry jako je šířka pásma, zpoždění a ztrátovost. Dále je možné vybrat si, zda zvolená úprava bude použita pouze v jednom směru nebo v obou směrech.

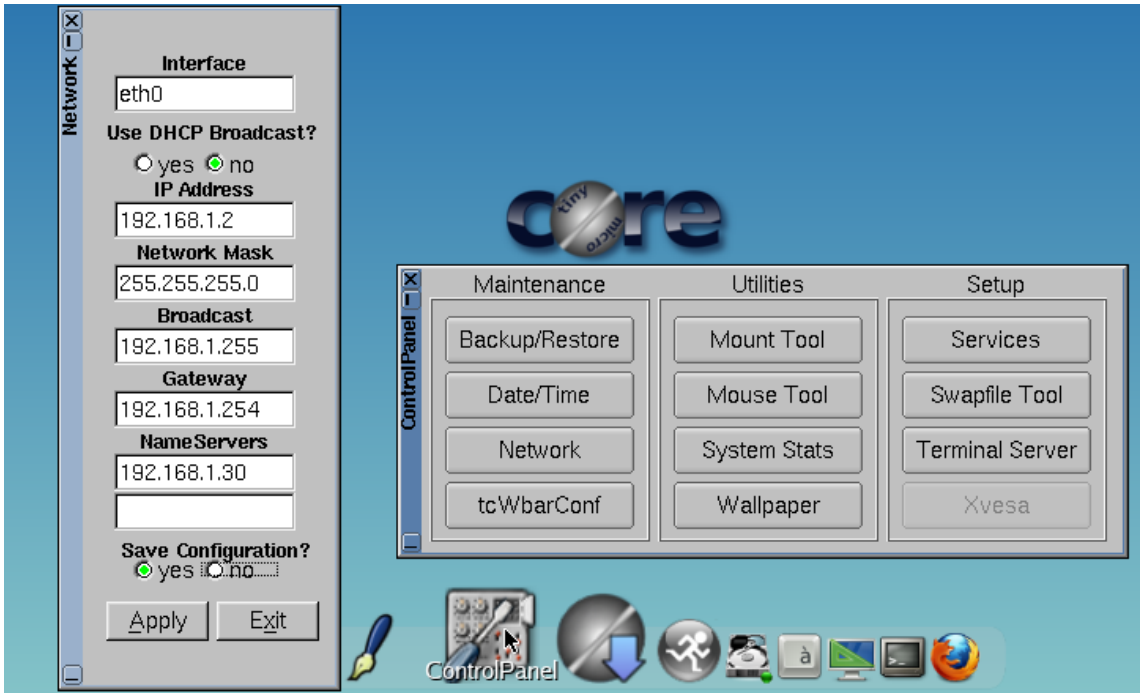


Obr. 4.2: Grafické prostředí NETem 0.4.

Pracovní stanice

Pracovní stanice bude mít za úkol stahovat daný soubor z vytvořených serverů FTP a TFTP a tím vyvolat komunikaci transportních protokolů TCP a UDP. Pro názorné ověření funkčnosti zapojení všech komponent vytvořeného testovacího prostředí, bylo vhodné vytvořit pracovní stanici s grafickým uživatelským rozhraním. Prostředí GNS nabízí virtualizace spousty typů klientských stanic různých operačních systémů. Pro požadavky laboratorní úlohy není nutné instalovat objemný operační systém, zabírající velké množství paměti. Proto byla vybrána instance GNS3 s názvem Firefox [38]. Tato instance běží na platformě TinyCore Linux. Nabízí grafické rozhraní a předinstalovaný plnohodnotný internetový prohlížeč Firefox.

Po instalaci a spuštění prvku Firefox je nutné nastavit síťové rozhraní. Studenti zde využijí i připraveného DNS serveru, který jim poslouží k překladu IP adres pro FTP a TFTP servery. K nastavení IP adresy je možné použít grafické konfigurační okno, které je možné zobrazit kliknutím na ikonu **Control Panel** a následném zvolením tlačítka **Network**. Ukázka správné konfigurace je na obrázku 4.3.



Obr. 4.3: Ukázka síťového nastavení klientské stanice PC.

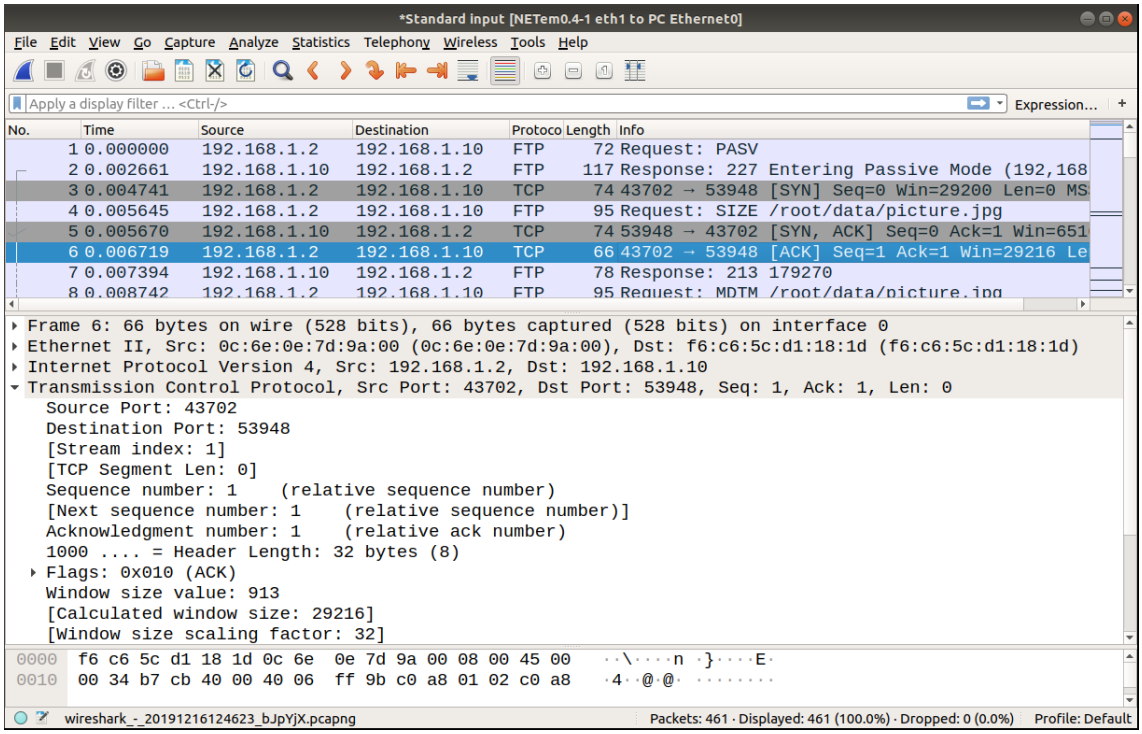
4.1.3 Popis analýzy síťové komunikace

Pro pochopení základních vlastností a rozdílů transportních protokolů TCP a UDP bude využito analýzy zachycených paketů při stahování souboru ze serverů FTP a TFTP. Studenti se zaměří především na analýzu navazování spojení protokolu

TCP v porovnáním s protokolem UDP. Dále porovnají hlavičky transportních protokolů a také přenosové techniky. Rozdílné přenosové techniky protokolu TCP a UDP budou dobře patrné v situaci, kdy studenti omezí šířku pásma a nastaví chybovost přenosového kanálu.

Program Wireshark

Jak již bylo zmíněno v kapitole 3.3.1, program Wireshark dokáže zachytit síťovou komunikaci a následně detailně zkoumat zachycená data. V našem případě je velkou výhodou fakt, že plnohodnotný programu Wireshark je již součástí instalačního balíčku GNS3 a s testovacím prostředím GNS3 je plně kompatibilní. Zakomponovaný program Wireshark tedy není nutné nijak instalovat a nastavovat. V rámci simulací je tedy možné zachytávání paketů kdykoliv zapnout a zachycená data analyzovat v reálném čase, nebo si je uložit pro případnou pozdější analýzu. Ukázka zachycené komunikace pomocí programu Wireshark se nachází na obrázku 4.4.



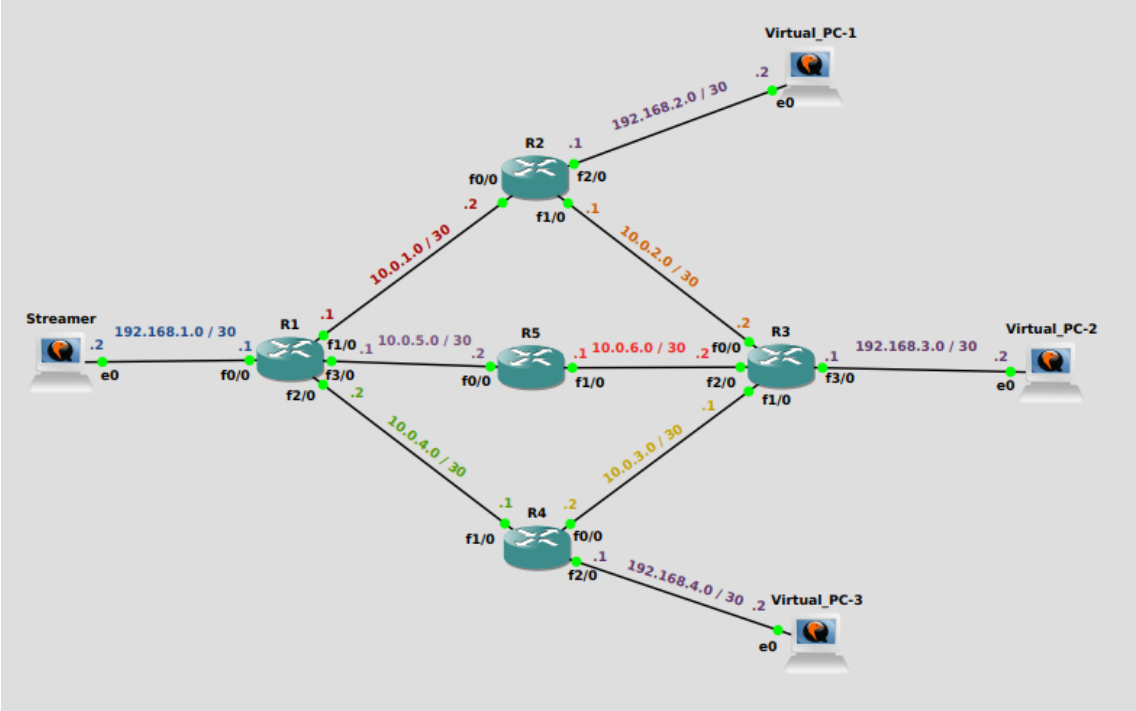
Obr. 4.4: Ukázka zachycené komunikace pomocí programu Wireshark.

4.2 Laboratorní úloha: Skupinové vysílání multicast

4.2.1 Popis laboratorní úlohy

Hlavním úkolem laboratorní úlohy je seznámit studenty s principem přenosu dat typu multicast. Multicastové vysílání bude realizováno v testovacím prostředí GNS3 na vytvořené síti pěti směrovačů, ke kterým budou připojené klientské stanice. Princip skupinového vysílání bude studentům předveden pomocí živého vysílání videa na vybranou skupinovou adresu. V rámci laboratorní úlohy bude realizováno také vysílání typu unicast, aby bylo možné demonstrovat hlavní vlastnosti a rozdíly skupinového vysílání.

Při správné konfiguraci laboratorní úlohy se budou moci jednotlivé klientské stanice připojit na vybranou skupinovou adresu a tím pádem budou moci generované živé vysílání videa zachytit a zobrazit. Skupinové vysílání bude realizováno pomocí protokolu PIM a to ve variantách Sparse a Dense Mode. Studenti budou mít za úkol obě varianty skupinové vysílání analyzovat pomocí programu Wireshark a následně umět definovat rozdíly mezi oběma variantami. Výsledné zapojení laboratorní úlohy v prostředí GNS3 se nachází na obrázku 4.5.

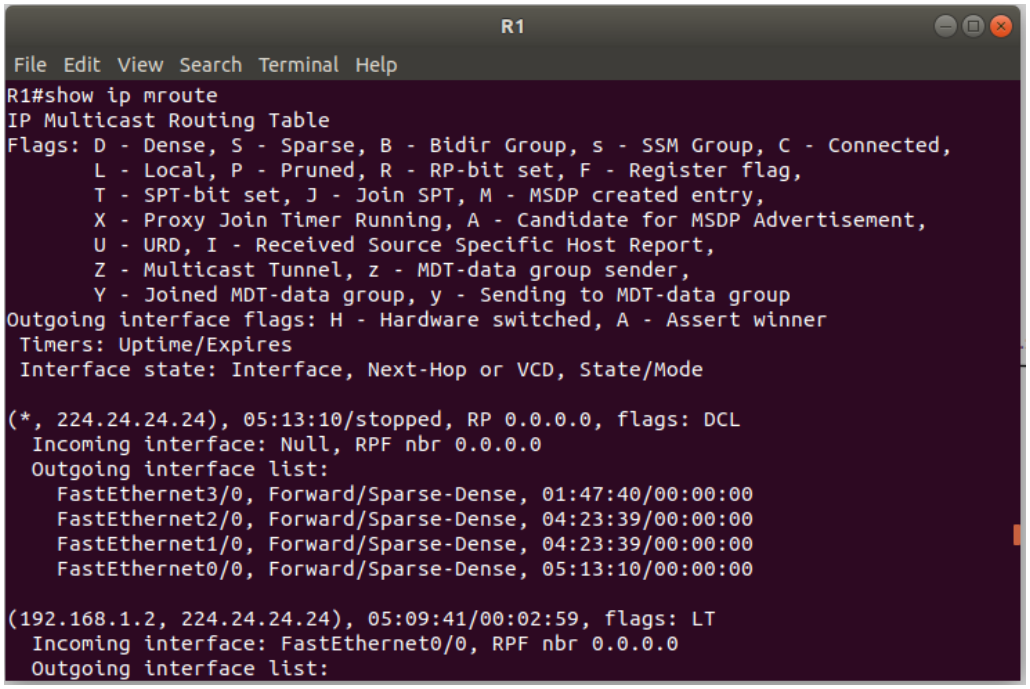


Obr. 4.5: Schéma zapojení pro demonstraci skupinového vysílání.

4.2.2 Konfigurace použitých síťových prvků

Směrovač Cisco 3600

Pro funkci směrovače byla vybrána virtualizace směrovače Cisco 3600 běžící na operačním systému IOS. Zvolená verze směrovače je pro testovací prostředí GNS3 volně dostupná a instalaci lze zdarma stáhnout ze serverů GNS3. Po importu a instalaci vybrané instance je směrovač k dispozici pod názvem c3600 v kategorii GNS3 Routers. Studenti pak mohou zvolený směrovač použít přesunutím vybrané instance do pracovní plochy testovacího prostředí GNS3. Konfigurace směrovače probíhá spuštěním konzole příkazového řádku pomocí simulovaného připojení Telnet (viz obrázek 4.6).



Obr. 4.6: Ukázka simulovaného telnet připojení směrovače Cisco 3600.

Pro správné fungování skupinového vysílání je nutné na všech směrovačích povolit skupinové vysílání příkazem:

```
R1(config)#ip multicast-routing
```

Dále je nutné u všech směrovačů provést konfiguraci všech aktivních síťových rozhraní a to nejdříve nastavením odpovídající IPv4 adresy a síťové masky příkazy:

```
R1(config)#int f0/0
R1(config-if)#ip add 192.168.1.1 255.255.255.252
```

Pro směrování skupinového vysílání bude využito protokolu PIM. Na všech aktivních síťových rozhraní je tedy nutné definovat tento protokol a vybranou variantu protokolu PIM. V rámci laboratorní úlohy je zvolena varianta SDM (Sparse Dense

Mode), která poskytuje výhodu v tom, že je možné operativně měnit varianty skupinového směrování mezi variantou Sparse a Dense jen podle toho, zda je definován RP (Rendezvous Point). Pokud není žádný směrovač v sítí definován jako RP, funguje skupinové směrování na principu Dense Mode. V opačném případě je směrování řízeno na principu Sparse Mode. Konfigurace protokolu PIM ve zvolené variantě je provedena příkazem:

```
R1(config-if)#ip pim sparse-dense-mode
```

Definování RP směrovače a zároveň vybrání varianty Sparse Mode je možné zadáním příkazu:

```
R1(config-if)#ip pim rp-address 10.0.5.2
```

Aby bylo možné v rámci laboratorní úlohy demonstrovat unicastové vysílání, je nutné na všech směrovačích definovat i klasický směrovací protokol, který zaručí, že všechny síťové prvky budou mezi sebou dostupné. V tomto případě je použit směrovací protokol OSPF. Na jednotlivých směrovačích je poté nutné definovat všechny přidružené sítě. Definování sítí je provedeno IPv4 adresou sítě, příslušnou wildcard maskou a definováním čísla oblasti. Ukázka konfigurace protokolu OSPF na směrovači R1:

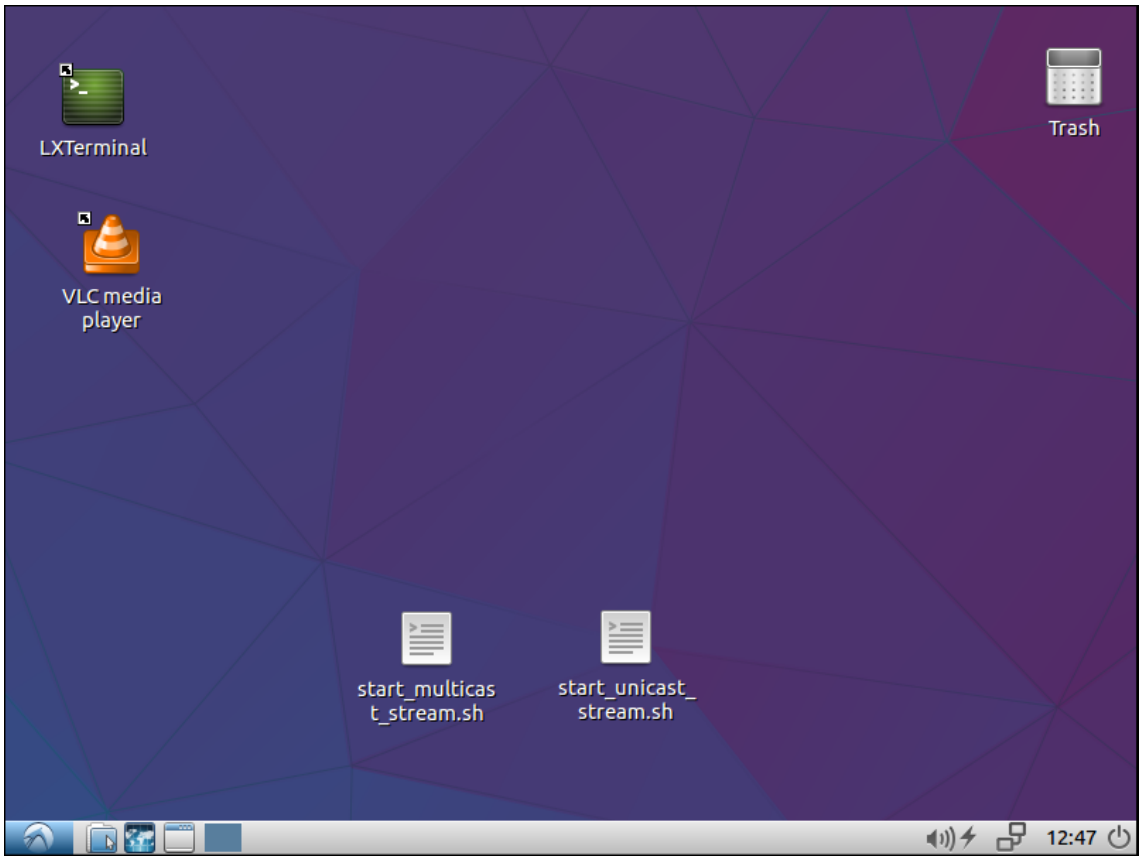
```
R1(config)#router ospf 1
R1(config-router)#network 192.168.1.0 0.0.0.3 area 0
R1(config-router)#network 10.0.1.0 0.0.0.3 area 0
R1(config-router)#network 10.0.5.0 0.0.0.3 area 0
R1(config-router)#network 10.0.4.0 0.0.0.3 area 0
```

Klientské stanice

Klientská stanice bude sloužit k vysílání a zachycení přenosu videa v reálném čase přes vytvořenou síť několika routerů. K tomuto úkolu byla vytvořena instance s názvem Virtual_PC. Tato instance běží na operačním systému Lubuntu 16.04.3. Jedná se o zjednodušený linuxový operační systém s grafickým prostředí, který není tolik náročný na výkonové prostředky. Linuxová distribuce navíc poskytuje výhodu, že je volně dostupná a není tak nutné řešit licencování. Přihlašovací heslo je stejné jako název uživatelského účtu: student. Náhled na grafické prostředí vytvořené instance Virtual_PC je na obrázku 4.7.

Aby bylo možné v rámci laboratorní úlohy testovací video vysílat a následně také zachytávat, byl na instanci Virtual_PC doinstalován program VLC media player. Program VLC je volně dostupný na svých webových stránkách[1] a je kompatibilní

[1] https://www.videolan.org/vlc/index.cs.html



Obr. 4.7: Ukázka klientské stanice Virtual_PC.

s celou řadou operačních systémů. Aby byl usnadněn start vysílání testovacího videa, byly vytvořeny dva skripty, které automaticky spustí požadovaný proud videa pomocí aplikace VLC. Kód pro multicastové vysílání odesílá video s názvem multicast_stream.mp4 pomocí protokolu RTP (port 5004) na skupinovou IPv4 adresu 224.24.24.24.

Ukázka výše popsaného skriptu s názvem **start_multicast_stream.sh** se nachází níže:

```
#!/bin/sh
vlc //home/student/Videos/multicast_stream.mp4
:sout=#transcode{vcodec=h264,vb=800,height=480,
acodec=mp3,ab=128,channels=2,samplerate=44100}
:rtp{dst=224.24.24.24:5004,mux=ts}
:ttl=10 :sout-keep -loop
```

Druhý skript pojmenovaný názvem **start_unicast_stream.sh** má za úkol vysílat video s názvem unicast_stream.mp4 pomocí protokolu UDP (port 1234) na IPv4 adresu instance Virtual_PC-1 192.168.2.2. Ukázka vytvořeného kódu druhého

skriptu je k vidění níže:

```
#!/bin/sh
vlc //home/student/Videos/unicast_stream.mp4
:sout=#transcode{vcodec=h264,vb=800,height=480,
acodec=mp3,ab=128,channels=2,samplerate=44100}
:udp{dst=192.168.2.2:1234,mux=ts}
:ttl=10 :sout-keep -loop
```

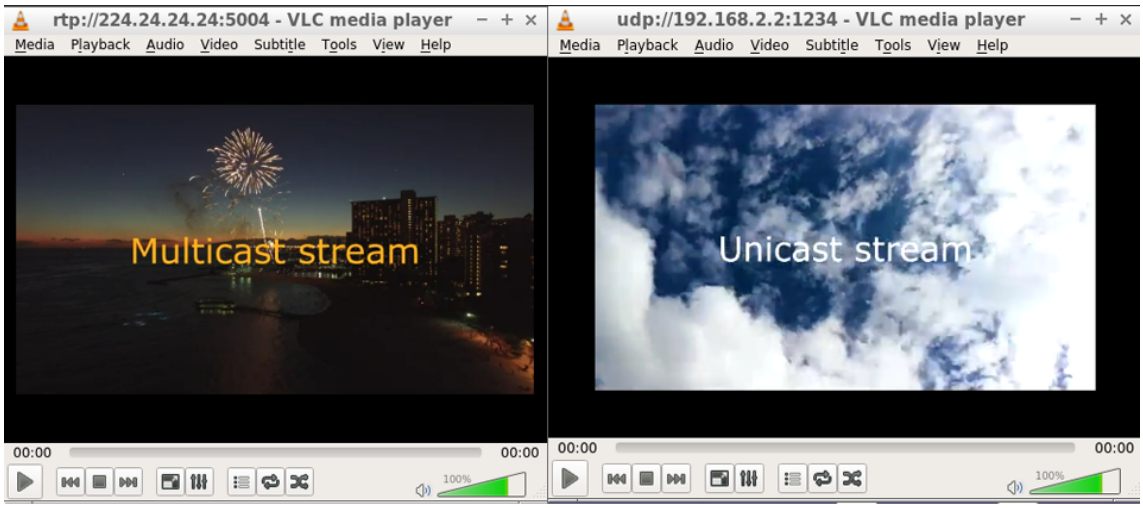
Oba skripty je možné spustit zároveň nezávisle na sobě. Pro ukončení vysílání je nutné vypnout okno aplikace VLC s příslušnou vysílací adresou. Pro ověření správného konfigurace a tím i ověření, zda oba typy vysílání fungují správně, je možné vysílané video zobrazit na příslušných klientských stanicích. To se provede spuštěním VLC playeru a kliknutím na položku *Media*, kde je nutní vybrat položku *Open Network Stream...* Pro zachycení skupinového vysílání (na instancích Virtual_PC-2 a Virtual_PC-3) je nutné zadat následující adresu vysílání:

`rtp://@224.24.24.24:5004`

Pro zachycení unicastového vysílání (na instanci Virtual_PC-1) je adresa:

`udp://@192.168.2.2:1234`

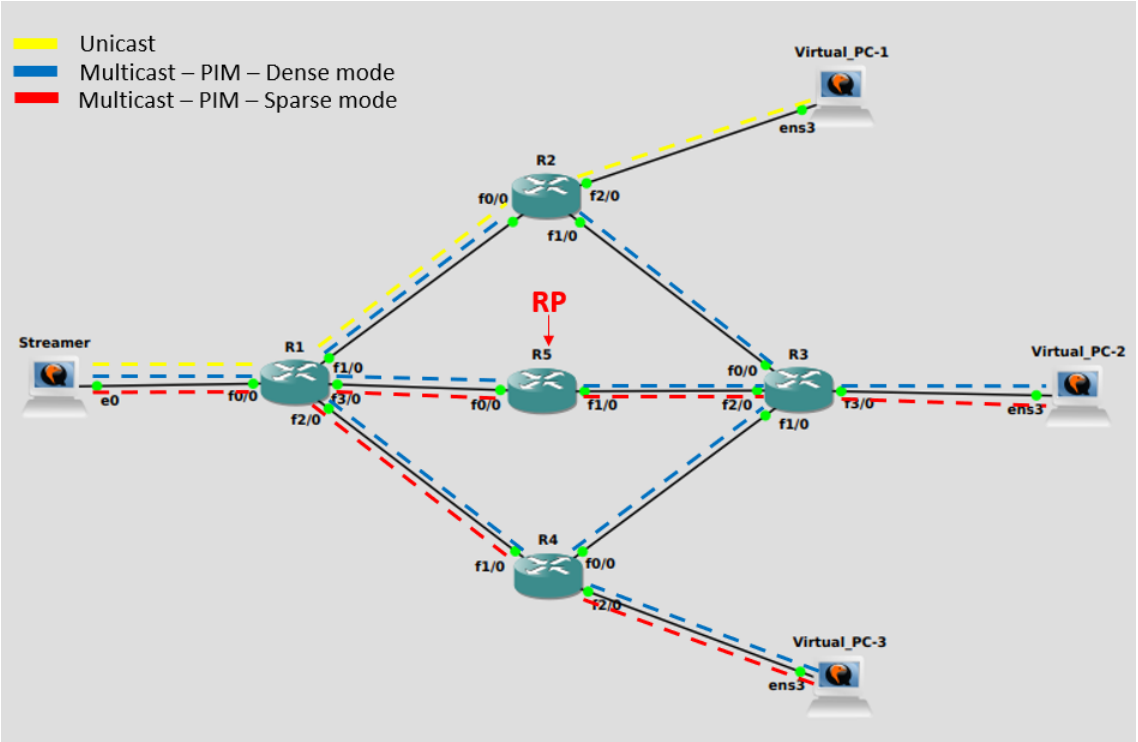
Při správné konfiguraci je možné z příslušných klientských stanic zobrazit generované vysílání z instance Streamer. Na obrázku 4.8 je k vidění správné zachycení unicast i multicast vysílání.



Obr. 4.8: Zobrazení unicast a multicast vysílání pomocí programu VLC.

4.2.3 Popis analýzy síťové komunikace

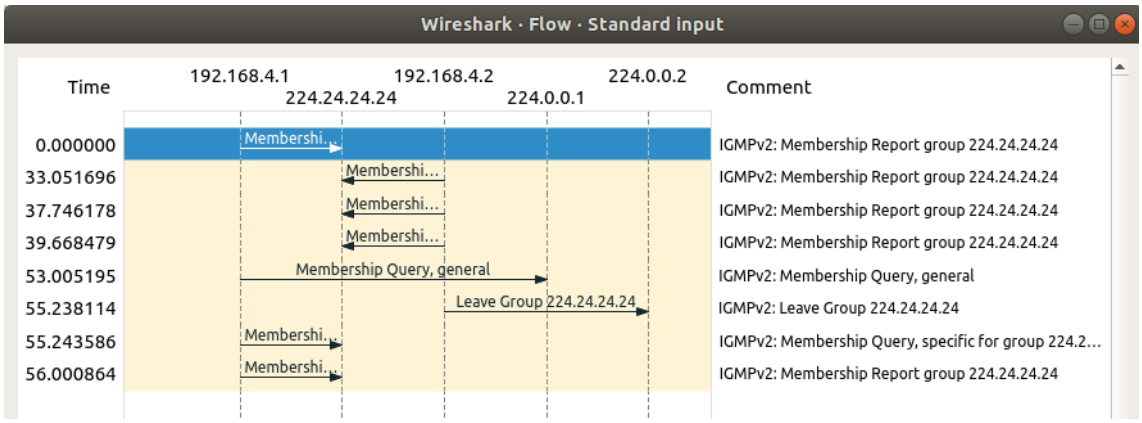
Jak již bylo uvedeno v části 4.2.1, po správné konfiguraci všech zadaných síťových prvků bude hlavním úkolem studentů provádět síťovou analýzu zadaných variant komunikace a to pomocí programu Wireshark, který je volně dostupný a je přímo součástí simulačního prostředí GNS3. Jedním z hlavních úkolů laboratorní úlohy je sestavit cestu komunikace ve 3 různých scénářích. Jednak je úkolem porovnat cestu komunikace vysílání typu Unicast oproti variantám skupinového vysílání při použití protokolu PIM ve Sparse a Dense modu. Studenti mají k dispozici i šablonu zadaného schéma laboratorní úlohy a očekává se od nich správné vyplnění cest jednotlivých variant. Správné vyplnění šablony cest jednotlivých se nachází na obrázku 4.9.



Obr. 4.9: Ukázka cesty jednotlivých variant vysílání.

Další variantou jakou je možné učit rozdíly mezi variantami PIM protokolu Sparse a Dense Mode je možnost zobrazení komunikace pomocí funkce Flow Graph aplikace Wireshark. Ta zobrazí zachycenou komunikaci v přehledných diagramech a zobrazí jednotlivé zprávy síťových protokolů i jednotlivé IP adresy, přes které zachycená komunikace probíhá. Analýzou protokolu IGMP, která probíhá mezi klientskými stanicemi a směrovači, je možné vidět komunikaci klientských stanic, žádající o přihlášení do zvolené skupinové adresy 224.24.24.24. Studenti mají za úkol zachytit komunikaci IGMP protokolu při spouštění živého vysílání, během sledování videa a také během ukončení skupinového vysílání, aby bylo patrné chování protokolu.

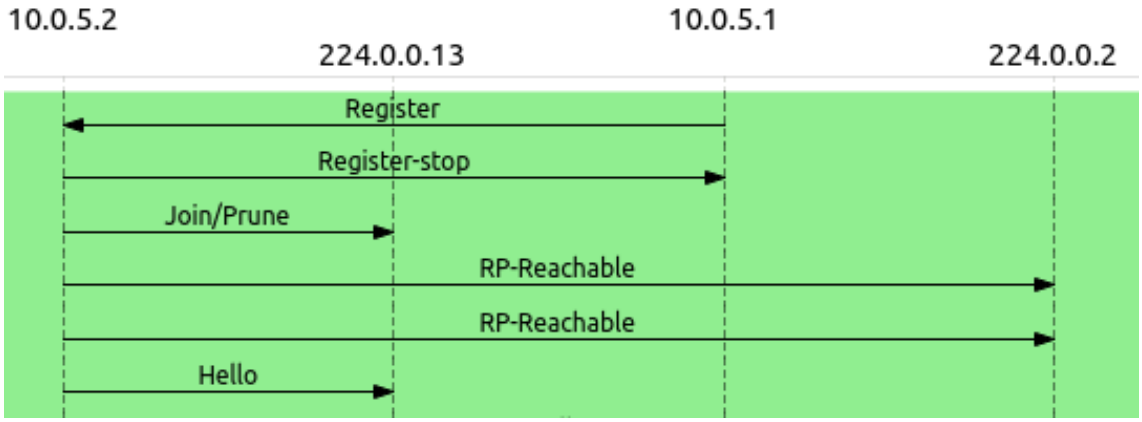
Na obrázku 4.10 jsou tak patrné vyhrazené IP adresy, které slouží k hlášení stavů (224.0.0.1) a k opouštění skupinových adres (224.0.0.2).



Obr. 4.10: Ukázka IGMP komunikace.

Při porovnání jednotlivých variant PIM protokolu jsou pomocí vývojového grafu patrné rozdílné zprávy, které slouží k různému principu směrování skupinové komunikace.

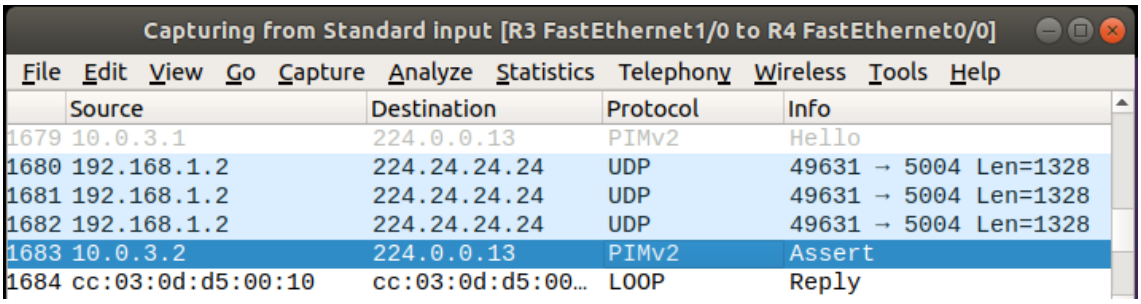
Ukázka zachycených zpráv protokolu PIM ve verzi Sparse Mode je k vidění na obrázku 4.11. Zachycené zprávy Register a Register-stop slouží k řízení skupinového vysílání přes definovaný RP směrovač. Tyto zprávy jsou typické právě pro variantu PIM Sparse Mode. RP směrovač po registraci zdroje vysílání přeposílá multicast komunikaci pouze do větví, které vedou přímo k zájemcům o skupinové vysílání.



Obr. 4.11: Zachycená komunikace PIM ve variantě Sparse Mode.

Komunikace protokolu PIM ve variantě Dense Mode, která naopak využívá postupného zaplavení všech definovaných PIM routerů. Při zahájení multicastového vysílání se generované pakety postupně zasílají mezi všechny PIM směrovače. Toto

vysílání se však později usměrní na nejkratší trasy, které vedou od cílových stanic, ke zdroji vysílání. Zachycená zpráva Assert je k vidění na obrázku 4.12.



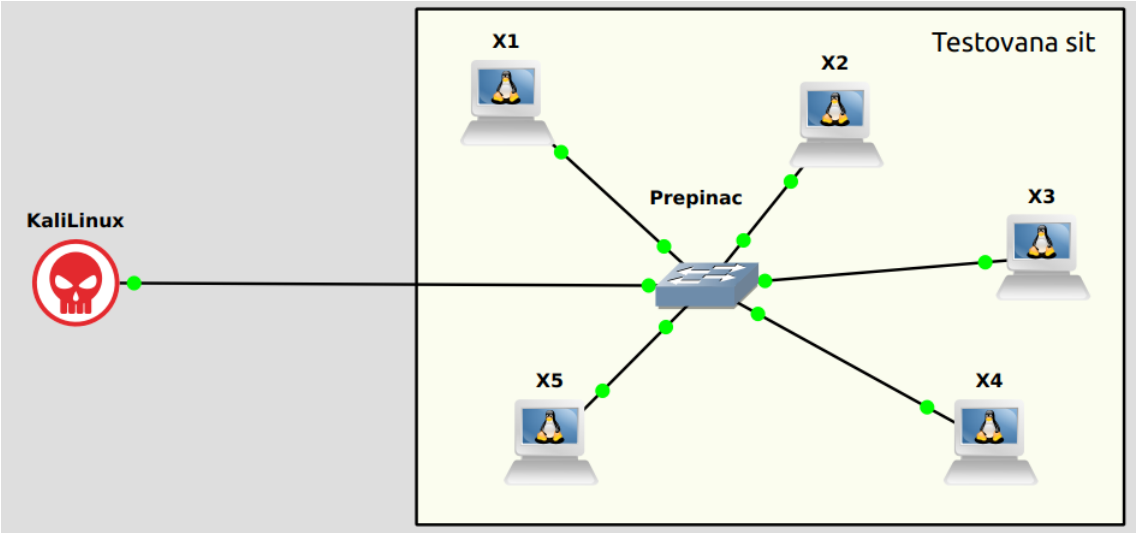
Obr. 4.12: Zachycená PIM zpráva Assert.

4.3 Laboratorní úloha: Základy penetračního testování

4.3.1 Popis laboratorní úlohy

Hlavním úkolem laboratorní úlohy je seznámit studenty se základy penetračního testování v simulačním prostředí GNS3. Pro potřeby penetračního testování byla vybrána volně dostupná virtualizace linuxové distribuce Kali Linux. Studenti si tak pomocí dostupných nástrojů v Kali Linux budou moci vyzkoušet základní skenování IP rozsahu, zjištění aktivních síťových zařízení, jejich IP adres a dostupných aplikačních portů. V prostředí GNS je nakonfigurována testovací síť s pěti různými instancemi. Jediná informace, kterou studenti dostanou k dispozici bude získaná IP adresa, kterou zjistí po připojení instance Kali Linux k centrálnímu přepínači z dosud skrytého DHCP serveru.

Postupným testování tak budou studenti získávat veškeré dostupné informace o dané síti a všech nakonfigurovaných síťových prvků. Postupně tak objeví servery WWW, FTP, DNS, SMTP a DHCP. Jedním z úkolů bude také prolomení bezpečnostního hesla k serveru FTP a odstavení služby HTTP serveru pomocí DoS útoku. Penetračním testováním si studenti ověří zranitelnost sítě s zařízeními v základním nastavení a bez jakýkoliv bezpečnostních prvků. Obrázek 4.13 reprezentuje ukázku schématu vytvořené laboratorní úlohy.

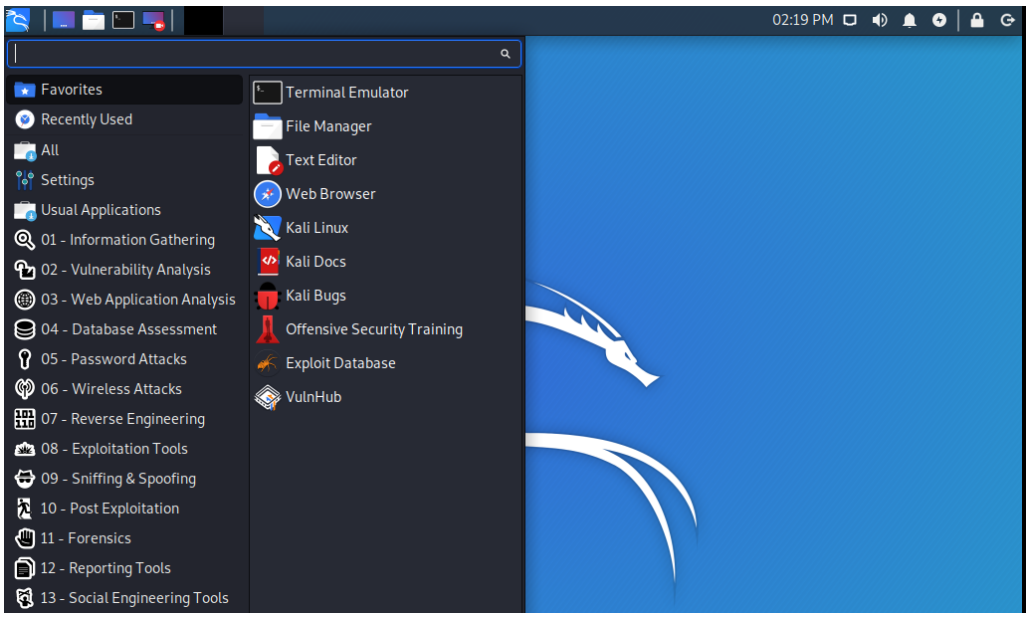


Obr. 4.13: Schéma zapojení pro demonstraci základů penetračního testování.

4.3.2 Konfigurace a popis použitých síťových prvků

Kali Linux

Kali Linux[39] je volně dostupná linuxová distribuce, která obsahuje spoustu nástrojů sloužící k tvorbě bezpečnostních analýz sítí a systémů a také k tzv. penetračnímu testování. V rámci laboratorní úlohy bude použita nejnovější verze z března roku 2020 označená jako Kali Linux 2020.1a. Ukázka prostředí virtualizace Kali Linux je na obrázku 4.14.



Obr. 4.14: Ukázka prostředí Kali Linux.

Do simulačního nástroje GNS3 byla virtualizace Kali Linux naimportována pod názvem instance **KaliLinux**, která je emulována pomocí systému Qemu. Studenti spustí vytvořený projekt s názvem **Pentest**, který má na pracovní ploše již umístěné a nakonfigurované instance síťových zařízení, jež budou studenti analyzovat. Do této pracovní plochy instanci KaliLinux přesunou a propojí ji s připraveným přepínačem (viz obrázek 4.13). Přihlašovací údaje instance KaliLinux jsou:

- Přihlašovací jméno: kali
- Heslo: student

Následně se instanci přiřadí IP adresa z DHCP serveru testované sítě a studenti tak po zjištění IP adresy mohou začít s penetračními testy dle zadaného manuálu.

X1 - DHCP server

DHCP server po připojení instance Kali Linux přiřadí IP adresu z definovaného rozsahu 192.168.1.10-20. Za pomoci zjištění přiřazené IP adresy a síťové masky bude prozrazen možný rozsah testovací sítě. Se zjištěným rozsahem budou studenti dále pracovat a budou postupovat ve zjišťování dalších informací.

Pro simulaci FTP serveru bude použit nástroj Toolbox [35], který lze volně stáhnout a importovat ze serveru GNS3. Jedná se o zjednodušenou verzi operačního systému Ubuntu, kde jsou nainstalované pouze základní komponenty, které slouží k vytváření jednoduchých serverů (WWW, FTP, TFTP, DHCP, Syslog a SNMP) v prostředí GNS3. Funkci DHCP serveru zde zajišťuje aplikace s názvem isc-dhcp-server. Ovládání a konfigurace Toolboxu je možné pomocí simulovaného připojení Telnet.

Pro konfiguraci DHCP serveru je nutné upravit soubor **/etc/dhcp/dhcpd.conf**. Níže uvedená konfigurace zajistí automatické přiřazení IP adres z rozsahu 192.168.1.10-192.168.1.20, síťové masky 255.225.255.0 a všesměrové adresy 192.168.1.255. Pro studenty skrytá IP adresa DHCP serveru je 192.168.1.3.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.20;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
default-lease-time 600;
max-lease-time 7200; }
```

Ovládání služby DHCP serveru (start, stop a restart) je možné příkazy:

```
service isc-dhcp-server start
service isc-dhcp-server stop
service isc-dhcp-server restart
```

Pokud služba DHCP serveru nejde spustit s tím, že je hlášeno, že instance stále běží, je nutné následujícím příkazem ukončit běžící proces:

```
rm /var/run/dhcpd.pid
```

X2 - FTP server

FTP server je realizován programem **vsftpd** v dostupné instanci Toolbox [35]. Skrytá IP adresa FTP serveru je 192.168.1.5. Soubory, které jsou dostupné po připojení na FTP server je nutné nahrát do složky /root dané instance GNS3. V našem případě se bude jednat o soubor report.pdf. Přihlašovací údaje FTP serveru, které se budou studenti snažit prolomit jsou následující:

- Přihlašovací jméno: root
- Heslo: gns3

Konfigurační soubor FTP serveru **/etc/vsftpd.conf** je upraven do následující podoby:

```
listen=YES #umožní samostatné spuštění služby vsftpd
anonymous_enable=NO #pro přístup k souborům je vyžadováno
    přihlášení
local_enable=YES #kontrola práv pro přístup na lokálním
   serveru
chroot_local_user=NO   #po přihlášení je konkrétním uži-
   vatelům umožněn přístup do soukromých složek
write_enable=YES #povolení ovládání pomocí FTP příkazů
local_umask=022 #nastavení plných práv lokálním uživatelů
    pro vytváření souborů
pam_service_name=vsftpd #název služby FTP
```

Ovládání služby FTP serveru (start, stop a restart) je možné příkazy:

```
/etc/init.d/vsftpd stop
/etc/init.d/vsftpd start
/etc/init.d/vsftpd restart
```

X3 - WWW server

I funkce HTTP serveru bude zajištěna pomocí instance Toolbox [35], kterou zprostředkovává předinstalovaná aplikace **nginx**. Skrytá IP adresa WWW serveru je 192.168.1.7.

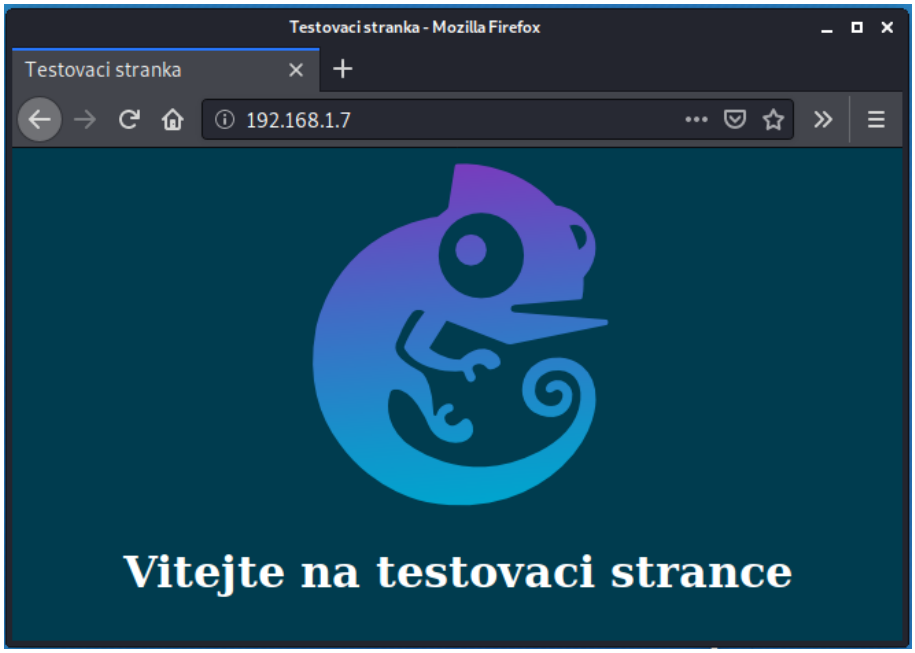
Konfigurační soubor webové stánky je uložen v souboru /var/www/html/index.html. HTML kód stránky je následující:

```
<!DOCTYPE html>
<html>
<head> <title>Testovaci stranka</title> </head>
<body style="background-color: #003c4f; color: white;">
<table border="0" cellspacing="0" cellpadding="0"
align="center"> <tbody> <tr>
<td style="text-align: center;"><img src="gns3.png" /></
   td>
</tr><tr>
<td>
<h1 style="text-align: center;">Vitejte na testovaci
   strance</h1>
</td></tr>
</tbody></table>
</body>
```

Ovládat službu nginx je možné příkazy:

```
/etc/init.d/nginx start
/etc/init.d/nginx stop
/etc/init.d/nginx restart
```

Ukázka vytvořené testovací stránky je na obrázku 4.15.



Obr. 4.15: Ukázka HTML stránky vytvořené na X3 (WWW serveru).

X4 - Mail server

Pro funkci poštovního serveru pracujícího s protokolem SMTP byl vybrán program **Postfix**, který byl nainstalován na linuxovou virtualizaci s operačním systémem Lubuntu 16.04.3 LTS. Tato virtualizace byla vložena do simulačního programu GNS3 pod názvem instance *lubuntu*. Programu Postfix je možné konfigurovat skrze konfigurační soubor **/etc/postfix/main.cf**, který byl upraven do následující podoby:

```
myhostname = mail.democloud.com

alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

myorigin = $mydomain
mydestination = $mydomain, localhost.$mydomain, localhost

relayhost =
mynetworks = 192.168.1.8/8
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
```

Výše uvedená konfigurace je dostačující, aby daný server naslouchal na portu TCP 25 protokolu SMTP. Službu Postfix pro poštovní server lze ovládat příkazy:

```
sudo systemctl start postfix
sudo systemctl stop postfix
sudo systemctl restart postfix
```

X5 - DNS server

Prostředí GNS3 nabízí možnost použití instance s názvem DNS [36], která má na základním linuxovém jádře OS Ubuntu předinstalovanou funkci **dnsmasq** zajišťující překlad IP adres. Po stažení a importu je nástroj DNS plně k dispozici. Po přetažení a spuštění zařízení je možné se připojit na konzoli pomocí připojení Telnet a začít s konfigurací. Pro požadavky laboratorní úlohy není nutné nikterak měnit konfiguraci virtuální stanice. Proto aby server naslouchal na otevřených portech protokolu DNS (tedy TCP 53 a UDP 53) stačí pouze spustit službu DNS serveru příkazy:

```
/etc/init.d/dnsmasq start
/etc/init.d/dnsmasq restart
```

4.3.3 Úlohy penetračního testování

Pomocí Kali Linux budou mít studenti za úkol níže uvedené hlavní body týkající se penetračního testování. V rámci bodů týkajících se skenování síťových prvků bude výstupem studentů vyplněná tabulka, která obsahuje souhrn zjištěných informací o zkoumaných prvcích X1-X5. Správně vyplněná tabulka 4.1 se nachází níže.

Tab. 4.1: Vyplněné informace zkoumaných síťových prvků.

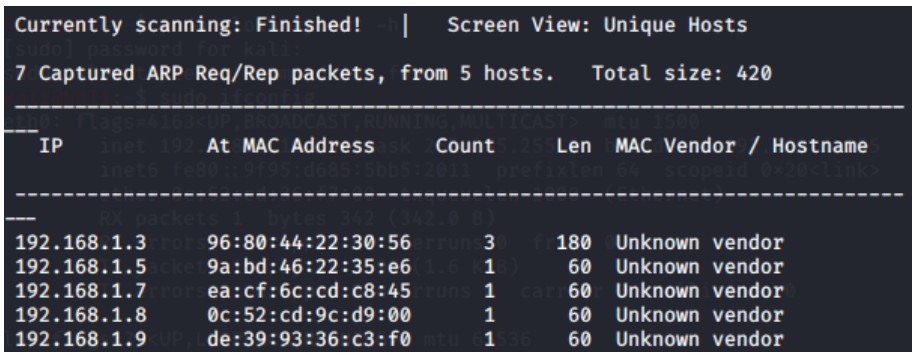
Server	IP adresa	MAC adresa	Otevřené TCP/UDP porty
X1	192.168.1.3	96:80.44:22:30:56	67-udp-open/filtered-DHCPs
X2	192.168.1.5	9A:BD:46:22:35:E6	21-tcp-open-FTP 69-udp-open/filtered-TFTP
X3	192.168.1.7	EA:CF:6C:CD:C8:45	80-tcp-open-HTTP
X4	192.168.1.8	0C:52:CD:9C:D9:00	25-tcp-open-SMTP
X5	192.168.1.9	DE:39:93:36:C3:F0	53-tcp-open-DOMAIN (DNS) 53-udp-open-DOMAIN (DNS)

Skenování aktivních síťových prvků

Pro zjištění aktivních síťových prvků bude využito nástroje **netdiscover**[40], jež používá informace z ARP tabulky, pro zjištění výpisu IP adres všech dostupných síťových zařízení. Studenti po spuštění aplikace terminál spustí vyhledávání pomocí příkazu:

```
sudo netdiscover -r 192.168.0.0/16
```

Na obrázku 4.16 je k vidění výstup aplikace, kde se nachází dostupné IP adresy ze zkoumané sítě. Tyto IP adresy studenti zapíší do předpřipravené tabulky a budou s nimi pracovat při vypracování dalších úkolů.



Obr. 4.16: Skenování aktivních síťových prvků.

Sken aktivních TCP portů

Skenováním aktivních TCP portů získaných IP adres bude napovězeno, které síťové aplikace běží na daných síťových zařízení. V rámci TCP skenování budou odhaleny používané síťové protokoly FTP, HTML, SMTP a DNS na příslušných IP adresách. Pro skenování síťových portů bude použit nástroj **nmap**[41]. V rámci tohoto úkolu bude použita metoda TCP SYN skenování. Na všechny síťové aplikační porty zvolené IP adresy budou zasílány zprávy typu SYN a od každého aktivního aplikačního portu se bude naslouchat a očekávat korektní odpověď [SYN,ACK] [41]. Níže jsou uvedeny příkazy, které budou studenti postupně zadávat.

```
sudo nmap -sT 192.168.1.3
sudo nmap -sT 192.168.1.5
sudo nmap -sT 192.168.1.7
sudo nmap -sT 192.168.1.8
sudo nmap -sT 192.168.1.9
```

Obrázek 4.17 obsahuje výstup aplikace nmap, který odhalil dostupných FTP port pro IP adresu 192.168.1.5.

```
kali@kali:~$ sudo nmap -sT 192.168.1.5
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-11 13:55 EDT
Nmap scan report for 192.168.1.5
Host is up (0.0024s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
MAC Address: 96:1B:A8:6E:AD:61 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 13.49 seconds
```

Obr. 4.17: Odhalení otevřeného FTP portu pro IP 192.168.1.5.

Dalším úkolem studentů bude analýza síťové komunikace pomocí aplikace Wireshark, kde je patrný postup skenování zvolené metody. Na obrázku 4.18 se nachází ukázka zachycené komunikace programem Wireshark, kde je patrná odpověď s příznaky [SYN, ACK] jež prozradili otevřený cílový port 21 protokolu FTP.

tcp.port == 21

Source	Destination	Protocol	Length	Info
192.168.1.10	192.168.1.5	TCP	60	64768 → 21 [SYN] Seq=0 Win=1024 Len=0
192.168.1.5	192.168.1.10	TCP	58	21 → 64768 [SYN, ACK] Seq=0 Ack=1 Win=
192.168.1.10	192.168.1.5	TCP	60	64768 → 21 [RST] Seq=1 Win=0 Len=0

Obr. 4.18: Zachycení korektní odpovědi k odhalení cílového portu 21.

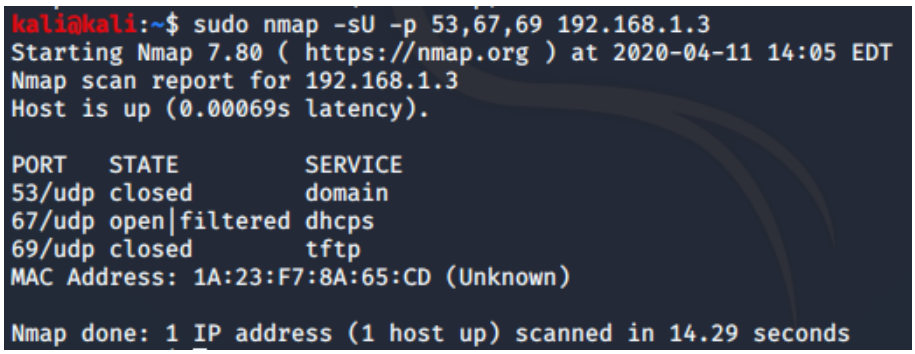
Sken aktivních UDP portů

Metoda UDP skenování je obtížnější a zdlouhavější v porovnání s předchozí metodou. Transportní protokol UDP neobsahuje metody třícestného navázání komunikace, tudíž není možné ověřit dostupný port na základě příznaků, které UDP protokol ani neobsahuje. Metoda UDP skenování proto odesílá prázdné UDP pakety na aplikační porty dané IP adresy a naslouchá odpovědi protokolu ICMP. Pokud dorazí odpověď ICPM typ 3 code 3 (Port unreachable), tak je jasné, že je cílový port nedostupný. V případě UDP protokolu je k ověření otevřeného UDP portu nutné, aby se provedlo kompletní a korektní navázání spojení [41]. Jelikož se často stane, že se kompletní komunikace nedokáže navázat, je nejčastější stav získané informace o UDP portu označován jako Open/filtered. Stav Open/filtered znamená, že nepřišla odpověď ICMP Port unreachable, ale také se nepodařilo programu nmap navázat korektní komunikaci, z důvodu například možného blokování firewallem [41].

Jak již bylo uvedeno, metoda skenování UDP je zdlouhavější a proto se studenti zaměří na získání informací o otevřených nebo filtrovaných portech 53 (pro DNS), 67 (pro DHCP) a 69 (pro TFTP). Skenování studenti provedou programem nmap a to příkazy [41]:

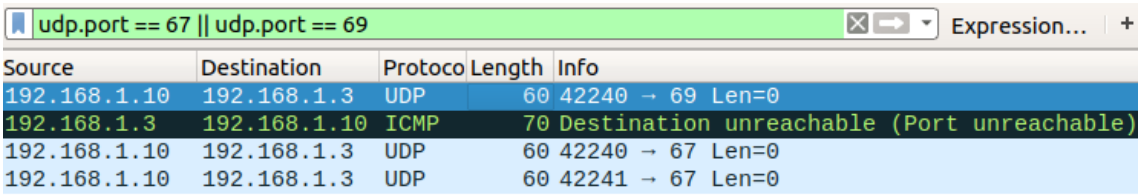
```
sudo nmap -sU -p 53,67,69 192.168.1.3
sudo nmap -sU -p 53,67,69 192.168.1.5
sudo nmap -sU -p 53,67,69 192.168.1.7
sudo nmap -sU -p 53,67,69 192.168.1.8
sudo nmap -sU -p 53,67,69 192.168.1.9
```

Výsledek skenování UDP portů (53,67 a 69) na IP 192.168.1.3 je na obrázku 4.19.



Obr. 4.19: Odhalení otevřeného/filtrovaného DHCP portu pro IP 192.168.1.3.

Ze zachycených dat programem Wireshark (viz obrázek 4.20) je patrné, že je správně označen port 69 jako uzavřený, jelikož pro tento port dorazila zpráva ICMP (Port unreachable). Pro port DHCP 67, tato zpráva nedorazila, a proto je označen jako otevřený/filtrovaný.



Obr. 4.20: Zachycení odpovědí cílového portu 67 a 69.

DoS útok na HTTP server

Pro možnost generovat DoS útoky je možné využít volně dostupného softwaru s názvem **SlowHTTPTest**[42]. Program SlowHTTPtest využívá zranitelnosti nezabezpečených HTTP serverů. Program odesílá na cílenou adresu data velmi nízkou rychlostí a po velmi malých částech. Při takovém průběhu se u nezabezpečeném serveru zaplní kapacity přístupu na HTTP stránku a po tuto dobu je cílený server nedostupný pro další požadavky.

Tato aplikace není součástí připravené Kali Linux distribuce a tak bylo nutné jej do této virtualizace doinstalovat příkazem:

```
sudo apt-get install slowhttptest
```

Program SlowHTTPtest lze po instalaci spustit pomocí příkazového řádku a patřičnými parametry. V případě laboratorní úlohy bude použit následující příkaz:

```
sudo slowhttptest -c 1000 -H -i 10 -r 200 -t GET
-l 60 -g -o /home/kali/vystup_testu -u http://192.168.1.7
```

Jednotlivé parametry znamenají následující [42]:

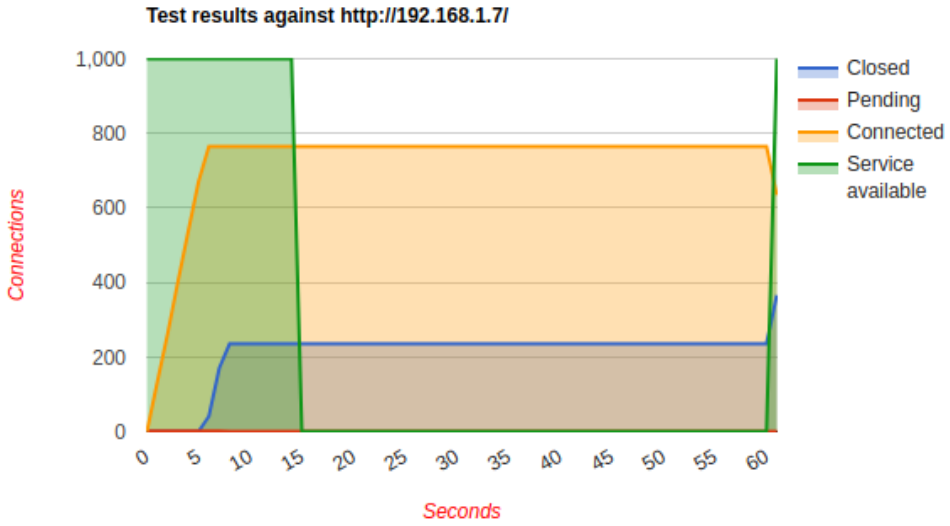
- -c 1000 (počet navázaných spojení),
- -H (typ útoků využívající nedokončené HTTP požadavky),
- -i 10 (interval mezi posíláním dat),
- -r 200 (definuje spojení za vteřinu),
- -t GET (typ odeslané zprávy),
- -l 60 (definuje délku trvání testu),
- -g -o vystup_testu (definuje název a cestu k finálnímu souboru, ve kterém lze zobrazit generovaný graf),
- -u http://192.168.1.7 (definice cílové adresy).

Při spuštění výše uvedeného příkazu začne program SlowHTTPtest generovat definovaný síťový provoz na HTTP adresu WWW serveru. Během průběhu programu jsou zobrazovány informace o počtu aktuálního spojení na daný server, statistiky v počtu uzavřených a navázaných spojení a především také status, jestli je daná služba stále dostupná nebo došlo k jejímu odstavení. Na obrázku 4.21 je vidět výstup z aplikace po dokončení testu. Je zde patrné, že probíhající test odstavil z provozu testovaný

```
Sun Apr 12 10:55:53 2020:
        slowhttptest version 1.6
 - https://code.google.com/p/slowhttptest/ -
test type:                        SLOW HEADERS
number of connections:            1000
URL:                              http://192.168.1.7/
verb:                             GET
Content-Length header value:      4096
follow up data max size:          68
interval between follow up data:  10 seconds
connections per seconds:          200
probe connection timeout:         5 seconds
test duration:                    60 seconds
using proxy:                      no proxy

Sun Apr 12 10:55:53 2020:
slow HTTP test status on 60th second:

initializing:         0
pending:              0
connected:            765
error:                0
closed:               235
service available:    NO
Sun Apr 12 10:55:54 2020:
Test ended on 61th second
Exit status: Hit test time limit
CSV report saved to //home/kali/vystup_test.csv
HTML report saved to //home/kali/vystup_test.html
```

Obr. 4.21: Ukázka analýzy programem SlowHTTPtest

WWW server. Studenti si také mohou ověřit, že během provádění testu, kdy se status (service available) změní na nedostupný, webové stránky opravdu nelze zobrazit pomocí webového prohlížeče.

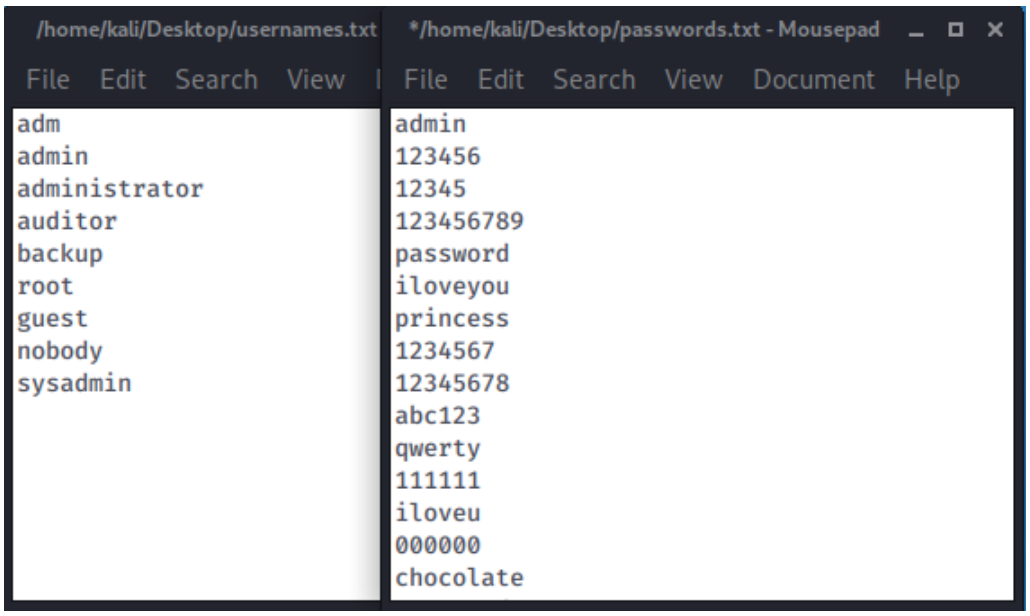


Obr. 4.22: Výsledný graf průběhu DoS útoku na WWW server.

Program SlowHTTPtest také reportuje cestu k definovanému souboru, kde se nacházejí souhrné informace výsledku testu, ale také především graf, dle kterého je patrné, že v čase 15 vteřin trvání testu se webová stránka stala nedostupnou. Výsledný graf, je k vidění na obrázku 4.22.

Prolomení přihlašovacích údajů FTP serveru

Jelikož vytvořený FTP server nemá nakonfigurované žádné bezpečnostní mechanismy, týkající se například zablokování přihlašovacích údajů po několika chybných pokusech o přihlášení, je možné na tento server použít metodu slovníkového útoku pro odhalení přihlašovacích údajů. K prolomení přihlašovacích údajů slovníkovým útokem je možné využít program **Hydra**[43], jež je mezi předinstalovanými programy virtualizace Kali Linux. K tomuto účelu byly vytvořeny dva textové soubory umístěné na ploše Kali Linux. Soubor *usernames.txt* obsahuje výpis 9 základních uživatelských účtů, které jsou velmi často používány, zatímco soubor *passwords.txt* obsahuje výpis hesel. Aby nebylo vyhledávání v rámci laboratorní úlohy příliš zdlouhavé byl soubor s hesly zredukován na 138 možných variant hesel. Ukázka výše zmíněných souborů je k vidění na obrázku 4.23.

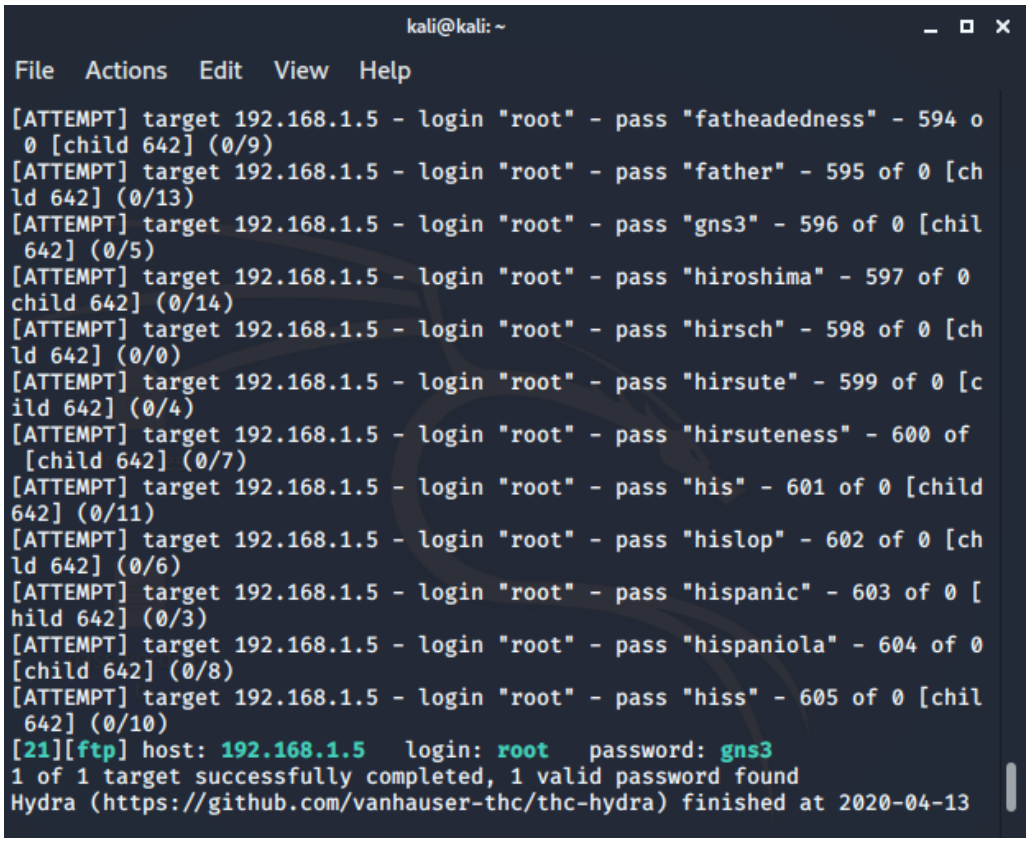


Obr. 4.23: Ukázka souborů obsahující přihlašovací údaje.

Získávání přihlašovacích údajů programem Hydra na testovaném FTP serveru je možné zadáním příkazu:

```
sudo hydra -L /home/kali/Desktop/usernames.txt -f
-P /home/kali/Desktop/passwords.txt ftp://192.168.1.5 -V
```

Po spuštění příkazu se začne program Hydra automaticky přihlašovat jednotlivými uživatelskými jmény a hesly, které bere vždy sestupně z připravených souborů. Testování je nastaveno tak aby studentům byl představeno postupné zkoušení jednotlivých přihlašovacích údajů a přibližně po 2 minutách testování jsou hledané přihlašovací údaje nalezeny. Odhalený výsledek programem Hydra je na obrázku 4.24.



Obr. 4.24: Výstup aplikace Hydra po nalezení hesla.

Po získání přihlašovacích údajů budou mít studenti za úkol přihlásit se na odhalený FTP server a zodpovědět na otázku, jejíž odpověď naleznou po otevření souboru report.pdf, který se nachází na příslušném FTP serveru.

5 Závěr

Na začátku diplomové práce byli čtenáři seznámeni s problematikou komunikace v počítačových sítí. Nejdříve bylo uvedeno rozdělení počítačových sítí do kategorií dle rozsáhlosti dané sítě, použité topologie a také dle dané hierarchie prvků. Poté byly představeny základní síťové prvky. V závěru první kapitoly práce je popsán referenční model ISO/OSI a architektura TCP/IP. Zde byly porovnány jejich základní vlastnosti a uvedeny hlavní rozdíly.

V další kapitole následoval popis základních protokolů TCP/IP rozdělených dle toho, na které vrstvě se vyskytují a kde pracují. Vybrány byly protokoly, jež byly použity v rámci vypracovaných laboratorních úloh. Z protokolů internetové vrstvy byly popsány protokoly IPv4, IGMP, PIM a ICMP. Při popisu ICMP protokolu byly popsány i programy PING a Traceroute, které se zaměřují na diagnostiku počítačových sítí. Dále byly uvedeny protokoly transportní vrstvy TCP a UDP a z protokolů aplikační vrstvy byly vybrány protokoly DNS, FTP a TFTP.

V praktické části byla představena vhodná testovací prostředí pro tvorbu laboratorních úloh. Vybráno bylo 6 testovacích prostředí. Konkrétně se jednalo o testovací prostředí Boson Network Simulator, IMUNES, EVE-NG, Netkit, NS-3 a GNS3. Pro tvorbu laboratorní úlohy byl, na základě porovnaných vlastností jednotlivých prostředí, zvolen nástroj GNS3. Vybrán byl především díky propracovanému grafickému prostředí, kvalitní dokumentaci a také díky tomu, že podporuje simulaci nepřeberného množství síťových prvků. Následoval popis instalace a přípravy vybraného simulačního prostředí GNS3, stejně tak byly uvedeny použité síťové nástroje.

V rámci praktické části práce byly vypracovány celkem tři laboratorní úlohy zabývající se sítovými technologiemi. První úloha má za úkol demonstrovat studentům základní rozdíly mezi transportními protokoly TCP a UDP. Druhá laboratorní úloha se věnuje skupinovému vysílání a třetí seznamuje studenty se základy penetračního testování. Kapitola obsahuje vždy popis hlavních cílů jednotlivých laboratorních úloh. Také je zde podrobně popsán popis přípravy všech síťových prvků daných úloh a ukázky požadovaných cílů a výstupů úloh. Všechny laboratorní úlohy budou studenti vypracovávat na připraveném testovacím prostředí GNS3. Časová náročnost všech úlohy byla modifikována přibližně na 1 a půl hodiny. Přenositelnost a funkčnost vytvořených virtualizací jednotlivých úloh byly prakticky otestovány na dvou odlišných PC. Vypracované návody pro studenty je možné nalézt v příloze práce. Vzorová řešení, stejně jako výstupy jednotlivých úloh, byly nahrány na přiložené DVD.

Literatura

[1] *What is a Network?* [online]. 2013 [cit. 14. 10. 2019]. Dostupné z:<`https://fcit.usf.edu/network/chap1/chap1.htm`>.

[2] PETERKA, Jiří. *eArchiv: Principy počítačových sítí* [online]. 1996 [cit. 14. 10. 2019]. Dostupné z: <`http://www.earchiv.cz/i_pri.php3`>

[3] DORDAL, L. Peter *An Introduction to Computer Networks* [online]. 2019 [cit. 14. 10. 2019]. Dostupné z: <`http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf`>

[4] *Personal Area Network* [online]. 2019 [cit. 14. 10. 2019]. Dostupné z: <`https://www.sciencedirect.com/topics/engineering/personal-area-network`>

[5] JEŘÁBEK, Jan. *Komunikační technologie.* Skriptum FEKT Vysoké učení technické v Brně, 2019. s. 1-175.

[6] GOYAL, Anshika *Types of area networks – LAN, MAN and WAN* [online]. 2018 [cit. 14. 10. 2019]. Dostupné z: <`https://www.geeksforgeeks.org/types-of-area-networks-lan-man-and-wan/`>

[7] *Computer Network Topology* [online]. 2019 [cit. 18. 10. 2019]. Dostupné z: <`https://computernetworktopology.com/`>

[8] KOTON, Jaroslav *Aktivní prvky datových sítí pro integrovanou výuku VUT a VŠB-TUO* Skriptum FEKT Vysoké učení technické v Brně, 2014. s. 1-175.

[9] FRED, Halsall. *Computer networking and the internet.* Fifth edition. Edinburg: Addison-Wesley, 2005, 803 s. ISBN 0-321-26358-8.

[10] CIGÁNEK, J. *Bezpečnostní analýza firewallu* [online]. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. 2017. s. 1-71.

[11] FOROUZAN, B. A. *TCP/IP Protocol Suite.* Fourt edition, Boston: McGraw-Hill Higher Education, 2010, 979 stran. ISBN 978-0-07-337604-2.

[12] BROOKSHEAR, J. Glenn. *Informatika* 1. vydání, Brno: Computer Press, 2013, 559 stran. ISBN 978-80-251-3805-2.

[13] JEŘÁBEK, Jan. *Pokročilé komunikační techniky* Skriptum FEKT Vysoké učení technické v Brně, 2018. s. 1-174.

[14] *IANA IPv4 Address Space Registry* [online]. 2019 [cit. 31. 11. 2019]. Dostupné z: <https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>

[15] *UDP Header Format* [online]. 2010 [cit. 26. 10. 2019]. Dostupné z: <https://www.inacon.de/ph/data/ICMP/ICMP-Message-Format_OS_RFC-792_.htm>

[16] *TRACEROUTE(8) System Manager's Manual* [online]. 2019 [cit. 3. 12. 2019]. Dostupné z: <https://man.openbsd.org/traceroute>

[17] *Internet Group Management Protocol, Version 2* [online]. 1997 [cit. 29. 2. 2020]. Dostupné z: <https://tools.ietf.org/html/rfc2236>

[18] *Internet Group Management Protocol, Version 3* [online]. 2002 [cit. 29. 2. 2020]. Dostupné z: <https://tools.ietf.org/html/rfc3376>

[19] *Protocol Independent Multicast - Dense Mode (PIM-DM)* [online]. 2005 [cit. 5. 4. 2020]. Dostupné z: <https://tools.ietf.org/html/rfc3973>

[20] *Protocol Independent Multicast - Sparse Mode (PIM-SM)* [online]. 2016 [cit. 5. 4. 2020]. Dostupné z: <https://tools.ietf.org/html/rfc7761>

[21] *UDP Header Format* [online]. 2010 [cit. 26. 10. 2019]. Dostupné z: <https://www.inacon.de/ph/data/UDP/UDP-Header-Format_OS_RFC-768.htm>

[22] *TCP Header Format* [online]. 2010 [cit. 26. 10. 2019]. Dostupné z: <https://www.inacon.de/ph/data/TCP/TCP-Header-Format_OS_RFC-793.htm>

[23] *FILE TRANSFER PROTOCOL (FTP)* [online]. 1985 [cit. 5. 4. 2020]. Dostupné z: <https://tools.ietf.org/html/rfc959>

[24] *THE TFTP PROTOCOL (REVISION 2)* [online]. 1992 [cit. 5. 4. 2020]. Dostupné z: <https://tools.ietf.org/html/rfc1350>

[25] *DNS Message Format* [online]. 2010 [cit. 24. 10. 2019]. Dostupné z: <https://www.inacon.de/ph/data/DNS/DNS-Message-Format_OS_RFC-1035.htm>

[26] *Boson Network Simulator* [online]. 2019 [cit. 27. 10. 2019]. Dostupné z:<https://www.boson.com/netsim-cisco-network-simulator?r=1>.

[27] *Integrated Multiprotocol Network Emulator/Simulator* [online]. 2004 [cit. 27. 10. 2019]. Dostupné z:<http://imunes.net/about>.

[28] *EVE-NG Professional Cookbook* [online]. 2019 [cit. 27. 10. 2019]. Dostupné z:<https://www.eve-ng.net/images/EVE-COOK-BOOK-latest.pdf>.

[29] *Netkit* [online]. 2018 [cit. 27. 10. 2019]. Dostupné z:<http://wiki.netkit.org/index.php/Main_Page>.

[30] *Ns-3 Tutorial* [online]. 2019 [cit. 28. 10. 2019]. Dostupné z:<https://www.nsnam.org/docs/release/3.30/tutorial/html/index.html>.

[31] *Why should you use GNS3?* [online]. 2019 [cit. 28. 10. 2019]. Dostupné z:<https://gns3.com/software>.

[32] *GNS3 Installation on Linux* [online]. 2019 [cit. 28. 10. 2019]. Dostupné z:<https://docs.gns3.com/1QXVIihk7dsOL7Xr7Bmz4zRzTsJ02wklfImGuHwTlaA4/index.html>.

[33] BULLOCK, Jessey. PARKER, T. Jeff. *Wireshark® for Security Professionals.* First edition, Indiana: John Wiley & Sons, Inc, 2017, 348 stran. ISBN 978-1-118-91821-0.

[34] *Wireshark User's Guide* [online]. 2019 [cit. 11. 11. 2019]. Dostupné z:<https://www.wireshark.org/docs/wsug_html_chunked/index.html>.

[35] *Toolbox appliance* [online]. 2019 [cit. 3. 11. 2019]. Dostupné z:<https://docs.gns3.com/appliances/net_toolbox.html>.

[36] *DNS appliance* [online]. 2019 [cit. 3. 11. 2019]. Dostupné z:<https://docs.gns3.com/appliances/dns.html#appliance_usage>.

[37] *NETem appliance* [online]. 2019 [cit. 5. 11. 2019]. Dostupné z:<https://docs.gns3.com/appliances/netem.html>.

[38] *Firefox appliance* [online]. 2019 [cit. 5. 11. 2019]. Dostupné z:<https://docs.gns3.com/appliances/firefox.html>.

[39] *What is Kali Linux?* [online]. 2011 [cit. 8. 4. 2020]. Dostupné z:<https://www.kali.org/docs/introduction/what-is-kali-linux/>.

[40] *Netdiscover - active/passive ARP reconnaissance tool* [online]. 2011 [cit. 10. 4. 2020]. Dostupné z:<https://manpages.debian.org/unstable/netdiscover/netdiscover.8.en.html>.

[41] *Nmap Network Scanning* [online]. 2011 [cit. 10. 4. 2020]. Dostupné z:<https://nmap.org/book/man.html#man-description>.

[42] *SlowHTTPTest Package Description* [online]. 2020 [cit. 12. 4. 2020]. Dostupné z:<https://tools.kali.org/stress-testing/slowhttptest>.

[43] *Hydra Package Description* [online]. 2020 [cit. 13. 4. 2020]. Dostupné z:<https://tools.kali.org/password-attacks/hydra>.

Seznam symbolů, veličin a zkratek

ACK	Acknowledgment
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
CIDR	Classless Inter-Domain Routing
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
DR	Designed Router
FR	Frame Relay
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
ISN	Initial Sequence Number
ISO	International Organization for Standardization
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPS	Intrusion Prevention System
IPSec	Internet Protocol security
LAN	Local Area Network
LLC	Logical Link Control
MAC	Media Access Control
MPLS	Multiprotocol Label Switching
MTU	Maximum Transmission Unit
NAT	Network Address Translation
OS	Operating System
OSPF	Open Shortest Path First
OSI	Open Systems Interconnection
PAN	Personal Area Network
PC	Personal Computer
PDF	Portable Document Format
PDU	Protocol Data Unit

PIM	Protocol Independent Multicast
RIP	Routing Information Protocol
RFC	Request for Comments
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RR	Resource Records
RTP	Real-time Transport Protocol
RTT	Round Trip Time
QoS	Quality of Service
SCTP	Stream Control Transmission Protocol
SDU	Service Data Unit
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network
WAN	Wide Area Network
WebUI	Web User Interfaces

Seznam příloh

A Laboratorní úloha: Porovnání transportních protokolů TCP a UDP

A.1 Zadání

- Seznamte se se simulačním prostředím GNS3.
- V simulačním prostředí GNS3 dle zadaného schématu nakonfigurujte síťové prvky tak, aby byla umožněna síťová komunikace všech zadaných serverů. Realizujte konfiguraci lokálního DNS. Funkčnost otestujte.
- Pomocí programu Wireshark zachyťte síťovou komunikaci při stahování testovacího souboru ze serverů FTP a TFTP na klientskou stanici. Zachycení paketů proveďte před a po upravení kvalitativních vlastností síťového propojení.
- Dle analýzy zachycené síťové komunikace porovnejte základní vlastnosti transportních protokolů TCP a UDP.

A.2 Teoretický úvod

Transportní vrstva modelu TCP/IP

Transportní vrstva se v rámci modelu TCP/IP nachází na pomyslném rozhraní mezi nižšími vrstvami, které zajišťují komunikaci mezi vzdálenými stanicemi a vrstvou aplikační, pracující již s konkrétními procesy aplikací. Z tohoto důvodu musí být protokoly transportní vrstvy schopny zajistit korektní doručení dat jednotlivým aplikacím a tím pádem tak určitým způsobem jednotlivé procesy aplikací od sebe rozlišit. K rozlišení jednotlivých procesů slouží 16 bitové čísla, nazývána jako porty. Port může být zdrojový, nebo cílový. V případě zdrojového portu se jedná o lokální proces, který zahajuje komunikaci se vzdáleným procesem, jež je reprezentován portem cílovým. Porty se dělí do tří hlavních skupin:

- Známé - číselný rozsah portů: 0 – 1023. Často používané a známé aplikace.
- Registrované - číselný rozsah portů: 1024 – 49151. Méně používané aplikace, které jsou organizací IANA registrované.
- Soukromé a dynamické - číselný rozsah portů: 49152 – 65535. Dynamicky přiřazované porty, které nejsou fixně spojovány s určitou aplikací a porty soukromé.

Příklady známých portů nejpoužívanějších aplikačních protokolů se nacházejí v tabulce A.1. Kombinaci IP adresy a daného portu nazýváme socket.

Hlavním úkolem transportních protokolů je poskytovat požadovanou službu konkrétním protokolům vyšší aplikační vrstvy. Požadavky na služby transportních pro-

Tab. A.1: Porty vybraných aplikačních protokolů.

Aplikační protokol	**Číslo portu**	**Transportní protokol**
FTP	20 / 21	TCP
TFTP	69	UDP
DNS	53	TCP/UDP
DHCP	67 / 68	UDP
HTTP	80	TCP
HTTPS	443	TCP/UDP

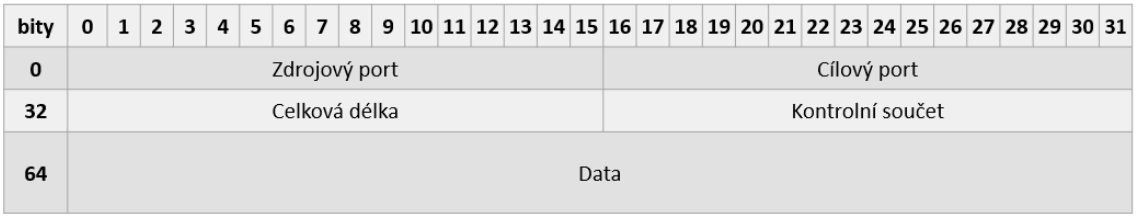
tokolů se mohou lišit zejména v požadované spolehlivosti přenosových služeb, nebo potřeby spojovaného či nespojovaného přenosu, které jsou odvozeny například dle požadované rychlosti předání dat. Dle požadovaných vlastností je pak konkrétním aplikačním protokolům vybrán vhodný protokol transportní vrstvy. Nedílnou součástí práce transportních protokolů je také segmentace dat v případech, že aplikace přenáší velké množství dat, jež je nutné rozdělit na několik menší oddílů. Před jednotlivé oddíly je poté přidáno identifikační záhlaví transportního protokolu a takto připravené jednotky jsou předány vrstvě síťové k následnému směrování. Provedení segmentace se však liší v závislosti na použitém transportním protokolu.

Nejpoužívanější protokoly transportní vrstvy jsou protokoly UDP a TCP. Příkladem dalších protokolů transportní vrstvy, které nejsou tak často využívány, jsou protokoly SCTP (Stream Control Transmission Protocol) nebo protokol RTP (Real-time Transport Protocol).

Protokol UDP

Protokol UDP (User Datagram Protocol) nabízí nespolehlivý a nespojovaný přenos. Jedná se o jednoduchý protokol, který nemá zakomponované žádné mechanismy řízení toku dat, kontrolu zahlcení nebo řízení chybových stavů.

Využití protokolu UDP dává smysl především v případech, kdy je požadován rychlý přenos krátkých zpráv, u kterých je akceptovatelné občasné selhání spojení. Typickým příkladem vyšších protokolů, kterým vyhovují vlastnosti UDP protokolu,



Obr. A.1: Záhlaví protokolu UDP.

jsou protokoly přenášející obraz a zvuk v reálném čase. O složitosti protokolu UDP vypovídá i jednoduché záhlaví (viz obrázek A.1). Základní jednotkou, se kterou protokol UDP pracuje se nazývá datagram.

Protokol TCP

TCP poskytuje aplikačním protokolům spojovaný a spolehlivý charakter přenosu aplikačních dat. Spolehlivost zaručují zakomponované mechanismy řízení toku dat, jež jsou schopné regulovat velikost přenášených dat tak, aby nedocházelo k zahlcení přenosového média a tím pádem ke ztrátě, či zahození přenášených dat. Rozdíl ve složitosti protokolu TCP, v porovnání s protokolem UDP je patrný dle rozsáhlejšího záhlaví, které je na obrázku A.2.

<table>
<tr><th>bity</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0</td><td colspan="16">Zdrojový port</td><td colspan="16">Cílový port</td></tr>
<tr><td>32</td><td colspan="32">Pořadové číslo odesílaného bajtu</td></tr>
<tr><td>64</td><td colspan="32">Pořadové číslo potvrzovaného bajtu</td></tr>
<tr><td>96</td><td colspan="4">Délka záhlaví</td><td colspan="6">Rezerva</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="16">Délka okna</td></tr>
<tr><td>128</td><td colspan="16">Kontrolní součet</td><td colspan="16">Ukazatel naléhavých dat</td></tr>
<tr><td>160</td><td colspan="32">Volitelné položky</td></tr>
<tr><td>192</td><td colspan="24">Volitelné položky (pokračování)</td><td colspan="8">Výplňkové bity do 32 bitů</td></tr>
<tr><td>224</td><td colspan="32">Data</td></tr>
</table>

Obr. A.2: Záhlaví protokolu TCP.

Základní jednotku, se kterou protokol TCP pracuje nazýváme segment. Před začátkem přenosu dat protokol TCP nejdříve naváže komunikaci (pomocí tzv. three-way handshake) a po dokončení přenosu dochází k ukončení spojení. Pro správný průběh komunikace protokol TCP využívá příznakových bitů, stejně tak se využívá i číslování odeslaných a potvrzených bajtů, či kontrolních součtů.

Pro protokol TCP najdou využití především aplikace, které si za žádných okolnosti nemohou dovolit jakoukoliv ztrátu dat při přenosu. TCP protokol využívají aplikační protokoly jako například protokol FTP, HTTP a SMTP.

Protokoly aplikační vrstvy TCP/IP:

Protokoly aplikační vrstvy tvoří procesy, které poskytují službu koncovým uživatelům. Aplikační vrstva má za úkol k zajištění korektní komunikace zprostředkovat spojení mezi daty aplikací a nižší transportní vrstvou.

Protokol FTP

Protokol FTP (File Transfer Protocol) poskytuje služby umožňující přenos souborů mezi síťovými prvky. Protokol FTP je typickým příkladem služby typu klient-server, který poněkud nestandardně využívá dvou odlišných spojení. Nejdříve je navázáno řídící spojení a až poté je umožněno spojení datové. Pro obě spojení se využívá spolehlivého transportního protokolu TCP. Řídící spojení komunikuje implicitně na portu 21 a zahajování tohoto spojení je vždy v kompetenci klienta. Navázané řídící spojení je udržováno po celou dobu komunikace. Datového spojení, využívá implicitně port 20 a bývá navazováno dynamicky v závislosti na aktuální potřebě přenášet konkrétní data, nebo při vykonávání zadaných příkazů.

Aby mohlo dojít ke stažení dat pomocí protokolu FTP, tak je vyžadováno přihlášení k vzdálenému FTP serveru. Přihlášení může být provedeno standardně pomocí přihlašovacího jména a zadáním příslušného hesla. Při povolení anonymního režimu je přihlášení možné jen pomocí zadání veřejně známého přihlašovacího jména a hesla. Protokol FTP nabízí pro práci se soubory i jednodušší uživatelské rozhraní, které je však plně dostačující pro všechny dostupné funkce FTP protokolu. Uživatelům je umožněno procházet adresářové struktury a mají přehled o souborech v daných složkách.

Protokol TFTP

TFTP (Trivial File Transfer Protocol) je velice zjednodušená verze předchozího protokolu FTP, jež se používá pouze pro přenos malých souborů, které dle základní specifikace nesmí být větší než 32 MB. Na rozdíl od FTP protokolu, se v případě TFTP protokolu používá nespojovaný a nespolehlivý přenos pomocí protokolu UDP. Standardně je protokolu TFTP vyhrazen port 69. Přenášená data jsou rozdělena do 39 bloků o velikosti 512 bajtů [1], která jsou postupně číslována a přenášena. Číslování jednotlivých bloků typicky začíná hodnotou 1. Po odeslání jednotlivého bloku musí dojít k potvrzení jeho doručení. Bez tohoto potvrzení tak není možné odeslat následující blok. Pokud nedojde k potvrzení daného bloku před vypršením daného limitu, musí dojít k novému odeslání nedoručeného bloku dat. Kontinuální potvrzování jednotlivých bloků dat má za následek také výslednou nižší přenosovou rychlost v porovnáním s protokolem FTP.

Zjednodušení protokolu se projevilo také na možnostech připojení k daným TFTP serverům. TFTP server nevyužívá žádného uživatelského přihlašování, stejně tak neumožňuje žádné uživatelské rozhraní umožňující například procházení adresářových

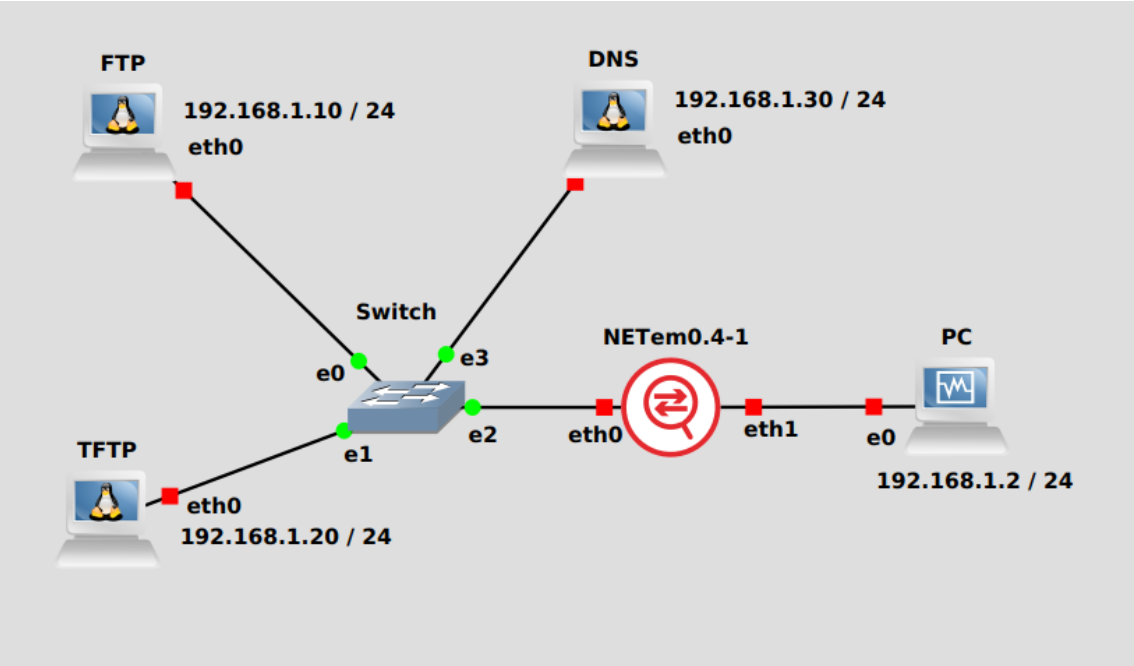
[1] Dle nejnovější specifikace TFTP protokolu (RFC 7440) z roku 2015 je možné navýšit maximální velikost bloků dat. Hraniční velikost by však neměla být vetší než je stanovená hodnota MTU, aby nedocházelo k fragmentaci na síťové vrstvě.

struktur. Stažení dat probíhá pomocí zadání příkazu v příkazové řádce, kde musí být přesně specifikována cesta s názvem požadovaného souboru a příslušným parametrem.

A.3 Vypracování

A.3.1 Příprava topologie v testovacím prostředí GNS3

Po spuštění virtualizace GNS3_LAB2 pomocí programu VMware spusťte simulační prostředí GNS3 a vytvořte nový projekt, který si libovolně pojmenujte. Heslo k uživatelskému účtu GNS3_LAB pro OS Ubuntu je: **student**. Dle níže uvedeného návodu nakonfigurujte a zprovozněte zapojení všech síťových prvků dle obrázku A.3.



Obr. A.3: Schéma zapojení laboratorní úlohy.

Konfigurace FTP serveru

Z nabídky všech předinstalovaných komponent (*Browse all device*) vyberte zařízení s názvem **Toolbox** a přetáhněte jej do pracovní plochy GNS3. Nástroj Toolbox je zjednodušený linuxový klient se základní verzí OS Ubuntu, který nám poslouží k základní konfiguraci FTP serveru. Název prvku v prostředí GNS3 je možné libovolně pojmenovat kliknutím pravého tlačítka myši na ikonu daného prvku a zvolením možnosti *Change hostname*. Pro konfiguraci FTP se připojte ke uživatelskému textovému rozhraní stroje pomocí připojení Telnet. Připojení k uživatelskému rozhraní zařízení

je možné provést kliknutím pravého tlačítka myši a zvolením možnosti *Console*. Před připojením k danému síťového prvku je nutné mít požadovaný prvek spuštěný. To lze provést vybráním možnosti *Start* u konkrétního prvku, nebo kliknutím na *Start/resume all nodes* na hlavním panelu, při kterém se spustí všechny prvky na pracovní ploše. Po úspěšném připojení k rozhraní Toolboxu se otevře nové okno s příkazovou řádkou. Nyní je možné postupně vkládat textové příkazy. Aby mohly ostatní simulované prvky komunikovat s FTP serverem je nutné nejdříve zadat IP adresu na dané fyzické rozhraní stroje. Příkazem:

```
ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
```

bude na na rozhraní eth0 nastavena požadovaná IP adresa. Parametrem *up* zadané fyzické rozhraní (eth0) zaktivujeme.

Pro základní konfiguraci, která bude v rámci laboratorní úlohy dostačující, upravte konfigurační soubor **/etc/vsftpd.conf** do následující podoby:

```
listen=YES #umožní samostatné spuštění služby vsftpd
anonymous_enable=NO #pro přístup k souborům je vyžadováno
    přihlášení
local_enable=YES #kontrola práv pro přístup
chroot_local_user=NO   #po přihlášení je konkrétním uži-
   vatelům umožněn přístup do soukromých složek
write_enable=YES #povolení ovládání pomocí FTP příkazů
local_umask=022 #nastavení plných práv lokálním uživatelů
pam_service_name=vsftpd #název služby FTP
```

Zápis nebo změnu konfiguračního souboru můžete provést zadáním příkazu:

```
nano /etc/vsftpd.conf
```

Po každé změně konfiguračního souboru nezapomeňte změny souboru uložit a také je nutné službu FTP serveru restartovat. Můžete tak učinit příkazy:

```
/etc/init.d/vsftpd stop
/etc/init.d/vsftpd start
/etc/init.d/vsftpd restart
```

Soubory, které budeme používat ke stažení z FTP serveru na klientskou stanici, je nutné nejdříve nahrát do složky **/root**. Pro otevření adresářové struktury FTP serveru je nutné kliknout pravým tlačítkem myši na ikonu daného zařízení a zvolit položku *Show in file manager*. Adresářová struktura daného zařízení v GNS3 se otevře v novém okně. V tomto okně vstupte do složky root (bude nutné zadat uživatelské heslo: student) a zkopírujte do ní testovací soubor picture.jpg, který je umístěný na ploše (ve složce *Desktop*) virtuální stanice, na které je aplikace GNS3

nainstalováno. Správné provedení kopírování je možné ověřit vypsáním obsahu adresáře */root* příkazem: *ls /root*.

Konfigurace TFTP serveru

Stejně jako u FTP serveru, tak i v případě vytvoření simulace TFTP serveru použijte zařízení **Toolbox**. Do pracovní plochy z nabídky dostupných zařízení přetáhněte zařízení Toolbox a pro lepší orientaci jej přejmenujte. Po připojení k textovému uživatelskému rozhraní TFTP serveru nastavte požadovanou IP adresu na rozhraním eth0 a toto rozhraní aktivujte.

Dále upravte konfigurační soubor pro TFTP server **/etc/default/tftpd-hpa** následovně:

```
TFTP_USERNAME="tftp"  #název služby TFTP
TFTP_DIRECTORY="/tftpboot" #složka, obsahující stahované
   soubory
TFTP_ADDRESS="192.168.1.20:69" #nastavení IP adresy a
   portu transportní vrstvy
TFTP_OPTIONS="--secure --create"    #nastavení umožnující
    přístup pouze ke konkrétnímu souboru, povolení nahrá-
   vání nových souborů
```

Pro úpravu textu použijte textový editor Nano. Upravený soubor nezapomeňte uložit. Podobně jako u FTP serveru je při změně konfigurace nutné restartovat služby TFTP serveru příkazy:

```
service tftpd-hpa stop
service tftpd-hpa start
service tftpd-hpa restart
```

Testovací soubor picture.jpg (umístěný na ploše), překopírujte stejným způsobem jako u předchozího FTP serveru s tím rozdílem, že cílová složka TFTP serveru nyní bude složka **/tftpboot**. Ověřte zda se ve složce **/tftpboot** nachází testovací soubor vypsáním obsahu daného adresáře pomocí příkazu.

Konfigurace DNS serveru

Pro seznámení se s protokolem DNS bude v testovacím prostředí vytvořený lokální DNS server, který bude umět překládat IP adresy FTP a TFTP serveru na zvolená doménová jména. Prostředí GNS3 nabízí možnost použití instance s názvem DNS, která má na základním linuxovém jádře OS Ubuntu předinstalovanou funkci dnsmasq, zajišťující překlad IP adres. Do pracovní plochy z nabídky dostupných zařízení přetáhněte zařízení DNS. Po spuštění zařízení je možné se připojit na konzoli

pomocí připojení Telnet a začít s konfigurací. IP adresu nastavte úpravou souboru **/etc/network/interfaces** do následující podoby:

```
auto eth0 #zpřístupnění fyzického portu eth0
iface eth0 inet static #na rozhraní eth0 povolí manuální
   nastavení pro adresy ipv4
address 192.168.1.30 #zadání ipv4 adresy
netmask 255.255.255.0 #zadání masky podsítě
up echo nameserver 192.168.1.30 > /etc/resolv.conf
#zadání IP adresy DNS serveru a zapsání do konf. souboru
```

Výše uvedenou konfigurací bude nastaveno síťové rozhraní DNS serveru. Poslední příkaz zaručí, že ostatní zařízení v síti mohou o překlad IP adres požádat na zadané IP adrese. Pro aktivaci síťového adaptéru je nyní nutné celou instanci DNS vypnout a zapnout. Jelikož se jedná o zjednodušenou verzi operačního systému není možné restart provést pomocí příkazu, ale je nutné v pracovní ploše GNS3 kliknout pravým tlačítkem myši na instanci GNS3, zvolit položky *Stop* a poté znovu spustit položkou *Start*.

Nyní můžete přistoupit k nastavení seznamu IP adres, kterým přiřaďte volitelná doménová jména. Seznam adres se nachází v souboru **/etc/hosts**, do kterého připište IP adresy FTP a TFTP serverů a vámi zvolená doménová jména ve formátu `[IP] [doménové jméno]`. V případě naší laboratorní úlohy bude konfigurace vypadat následovně:

```
192.168.1.10 muj_ftp
192.168.1.20 muj_tftp
```

Po úpravě je nutné soubor uložit a restartovat službu DNS serveru dnsmasq. Učinit tak můžete příkazy:

```
/etc/init.d/dnsmasq stop
/etc/init.d/dnsmasq start
/etc/init.d/dnsmasq restart
```

Konfigurace přepínače

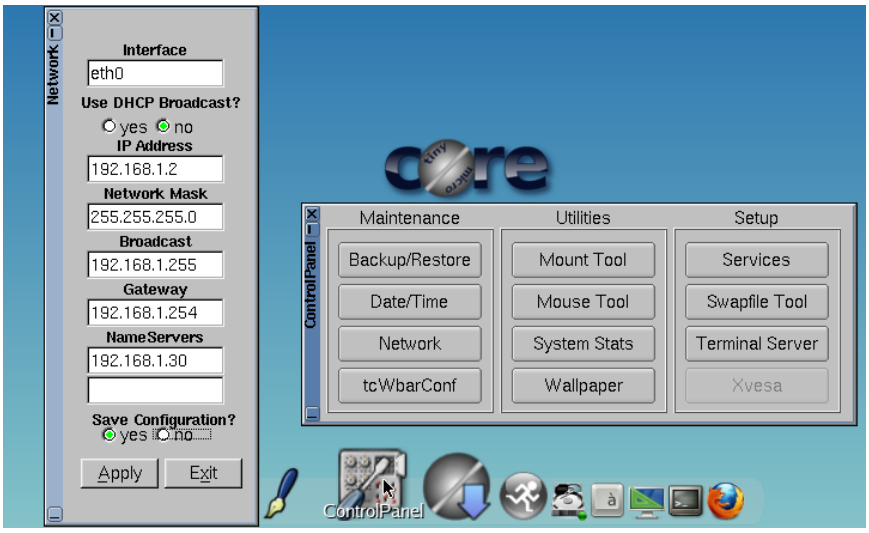
Nalezněte prvek přepínače **Ethernet switch** z nabídky předinstalovaných komponent a přetáhněte jej do pracovní plochy. Pro zajištění funkčnosti přepínače není nutná žádná dodatečná konfigurace stačí pouze připojit požadované prvky k volným fyzickým portům přepínače (Ethernet0-Ethernet7). Prvky je možné propojit pouhým kliknutím při spuštěném režimu *Add a link*. Připojte tedy vytvořený FTP server a TFTP server k volným portům přepínače.

Konfigurace prvku NETem 0.4

Pro regulaci kvalitativních parametrů přenosové linky v simulačním prostředí GNS3 lze použít instanci **NETem 0.4**, která dokáže nastavit požadovanou šířku pásma, zpoždění a ztrátovost. Dále je možné vybrat si, zda zvolenou úpravu použijete pouze v jednom směru, nebo v obou směrech. Přesuňte prvek **NETem 0.4** z nabídky na pracovní plochu a spusťte jej. Parametry lze upravovat v grafickém rozhraní, které se spustí po vybrání možnosti *Console* po kliknutí pravým tlačítkem myši na konkrétní prvek. Fyzický port eth0 zařízení NETem 0.4 propojte k volnému portu na přepínači dle zadaného schématu (obrázek A.3).

Konfigurace klientské stanice

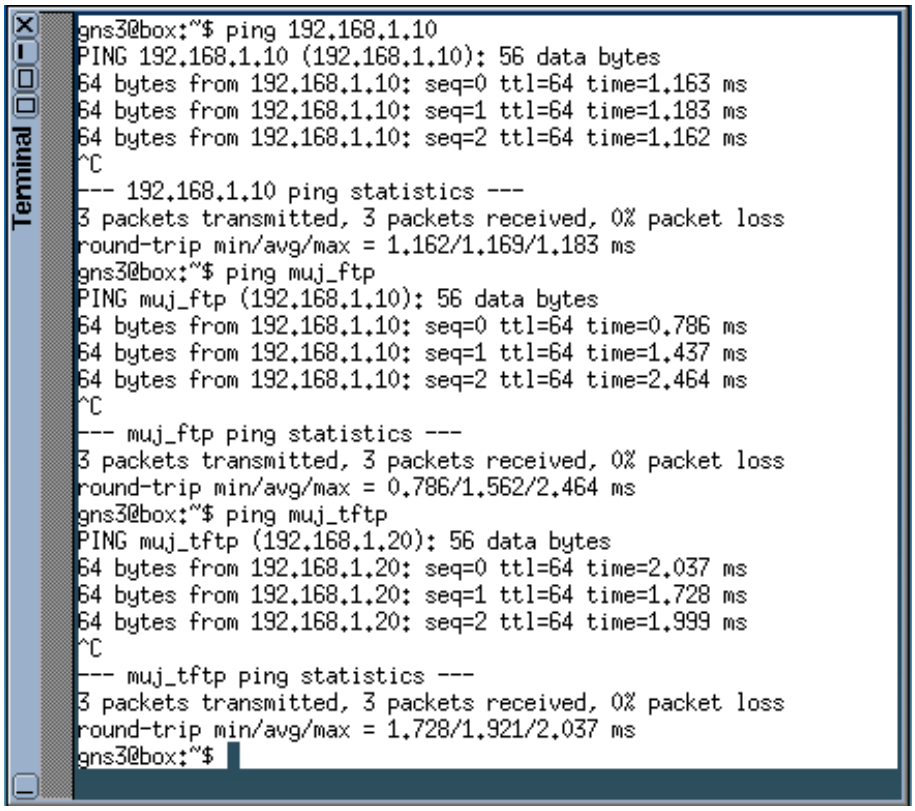
Abychom mohli názorně ověřit funkčnost zapojení všech komponent a přehledně řídit stahování souborů z vytvořených serverů FTP a TFTP v testovacím prostředí GNS3, bude vhodné použít pracovní stanici s grafickým uživatelským rozhraním. K tomuto účelu poslouží instance pojmenovaná **PC**. Jedná se o virtualizaci, která běží na platformě TinyCore Linux a obsahuje plnohodnotný internetový prohlížeč Firefox. Z nabídky dostupných zařízení vyberte instanci **PC** a přesuňte ji do pracovní plochy. Propojte síťové rozhraní Ethernet0 u instance PC s volným portem prvku NETem 0.4 viz obrázek A.3. Simulované PC spusťte a po připojení k virtuální konzoli prvku se zobrazí grafické rozhraní linuxového systému TinyCore Linux. Nyní nastavte IP adresu. Využijte zde i připraveného DNS serveru, který vám poslouží k překladu IP adres pro FTP a TFTP servery. K nastavení IP adresy použijte grafické konfigurační okno, které je možné zobrazit kliknutím na ikonu **Control Panel** a následném zvolením tlačítka **Network**.



Obr. A.4: Ukázka síťového nastavení klientské stanice PC.

Zde na rozhraní eth0 nastavte správnou IP adresu uživatelské stanice, masku a především také IP adresu DNS serveru (pole NameServers). Ukázka správné konfigurace je na obrázku A.4.

Nyní ověřte dostupnost všech vytvořených serverů. Z hlavní lišty klientské stanice vyberte aplikaci Terminal a postupným zadáváním příkazu: `ping <IP_adresa>` ověřte dostupnost FTP a TFTP serverů. Pomocí nástroje ping ověřte také správné fungování DNS serveru, kdy místo IP adresy zvolte vámi nastavené doménové jméno FTP a TFTP serveru. Ukázka správného nastavení prvků je na obrázku A.5, kde je vidět, že FTP server odpovídá na příkaz ping pomocí IP adresy i pomocí zvoleného doménového názvu.



Obr. A.5: Dostupnost FTP a TFTP serveru z klientské stanice PC.

Zahájení komunikace FTP a TFTP

Nyní přejdeme k realizaci samotné komunikace klientské stanice PC a serverů FTP a TFTP. Pro připojení k FTP serveru použijte internetový prohlížeč Firefox na klientské stanici PC. Po spuštění prohlížeče zadejte adresu FTP serveru ve formátu:

`ftp://muj_ftp`

Pokud je FTP správně nastaven zobrazí se přihlašovací okno k FTP serveru. Uživatelské jméno zadejte: **root** a heslo: **gns3**. Po správném přihlášení se zobrazí výpis nahraných souborů, kde se nachází soubor picture.jpg, který jste tam při předchozí konfiguraci FTP serveru nahráli.

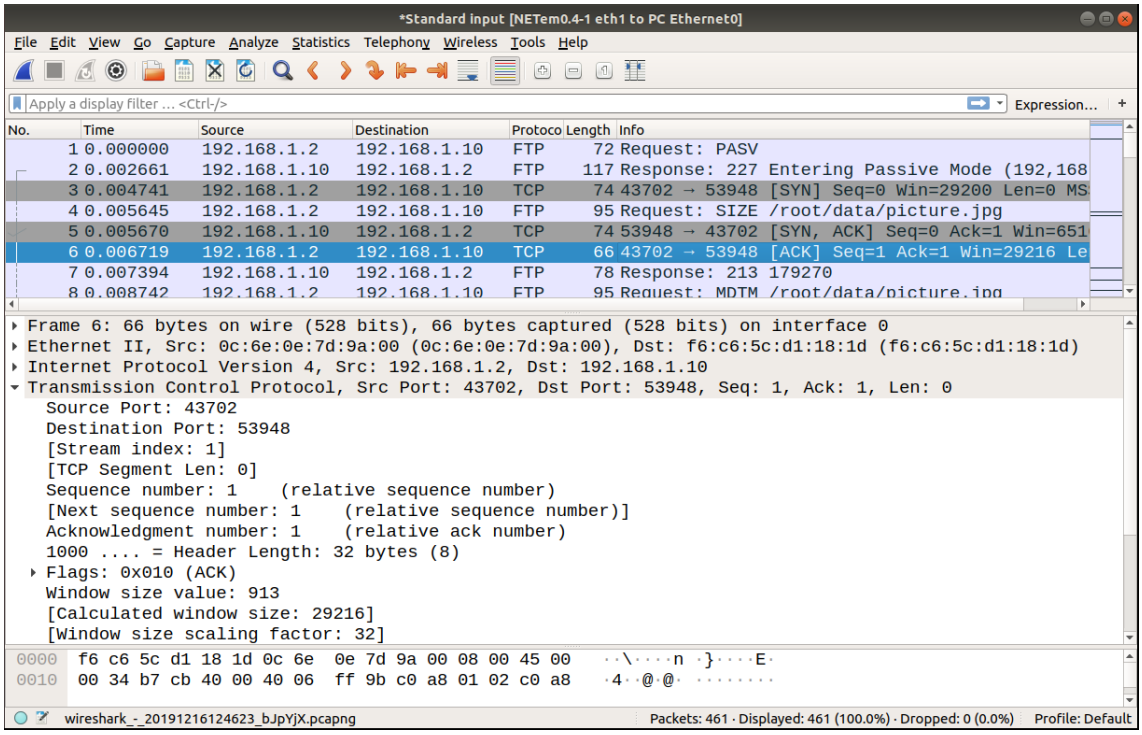
Jelikož je TFTP protokol v mnoha ohledech zjednodušenou verzí protokolu FTP, není možné se na TFTP server připojit stejným způsobem jako na server FTP. Pro připojení a stažení dat z TFTP serveru použijte aplikaci *Terminal*. Následujícím příkazem proveďte stažení požadovaného souboru:

```
tftp -g -r picture.jpg muj_tftp -b 1468
```

Parametrem -g se zajistí stahování vzdáleného souboru, který identifikuje parametr -r a název. Parametr -b slouží k nastavení maximální velikosti MTU, aby nedošlo k fragmentaci paketu. Pokud komunikace proběhla korektně, v aplikaci terminál se vypíší informace o stavu a trvání stažení požadovaného souboru.

A.3.2 Zachycení síťového provozu

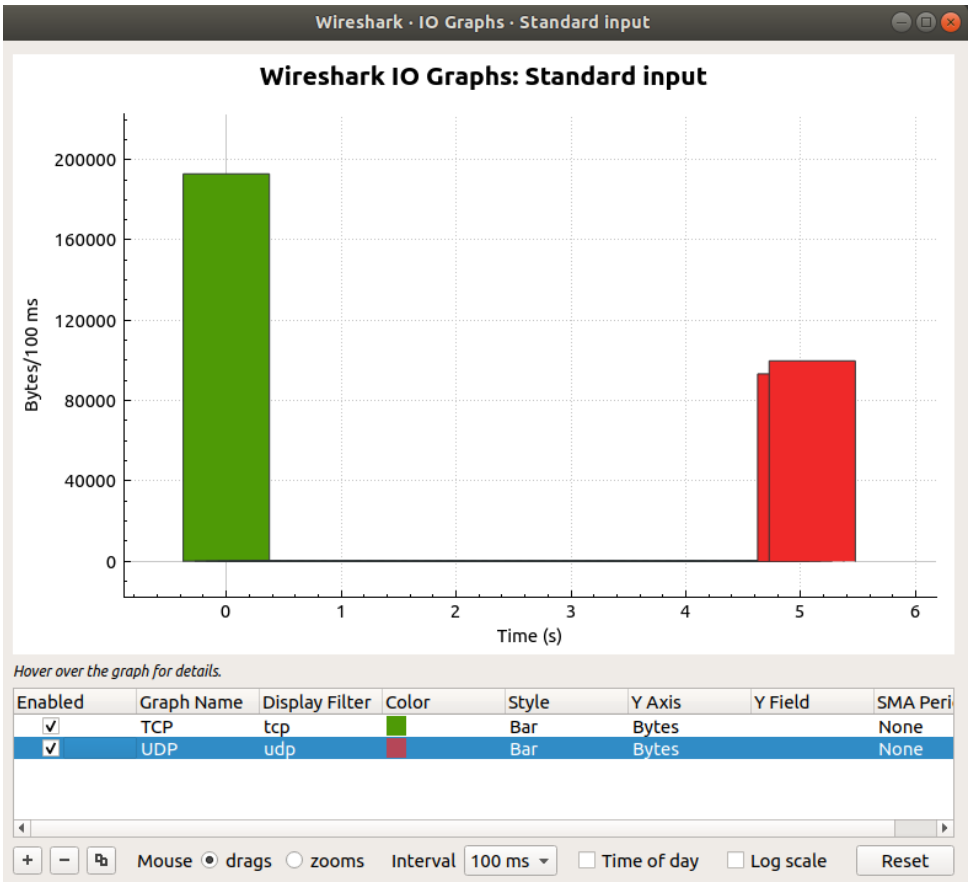
Pro zachycení síťového provozu použijte program Wireshark, který je již předinstalovaný v simulačním prostředí GNS3. Zachytávání paketů pomocí Wiresharku je možné spustit kliknutím pravého tlačítka myši na požadovaný síťový propoj a vybráním volby *Start capture*.



Obr. A.6: Ukázka správného zachycení protokolů FTP a TFTP.

Před automatickým spuštěním programu Wireshark je nutné určit název souboru, do kterého se budou zachycená data ukládat. Aktivní zachytávání síťové komunikace indikuje obrázek lupy na síťovém spojení. Pokud se aplikace Wireshark nespustí automaticky, je možné ji manuálně spustit výběrem možnosti *Start Wireshark* po stisku pravého tlačítka myši na vybraném spoji. Pro ukončení daného zachycení stiskněte na hlavním panelu programu Wireshark tlačítko *Stop capturing packets* a soubor se zachycenými daty si uložte ve formátu .pcapng pro pozdější analýzu. Pro další měření je nutné ukončit aplikaci Wireshark a také zastavit zachytávání paketů v GNS3 kliknutím pravého tlačítka myši na ikonu lupy (analyzovaný spoj) a vybraním možnosti *Stop capture*. Ukázka správného zachycení paketů je na obrázku A.6.

Jednou z používaných funkcí programu Wireshark je vytváření grafů **I/O Graph**, kterou najdete pod záložkou Statistics na hlavní liště. Tlačítkem plus a mínus lze přidávat, či odebírat množství generovaných a vykreslovaných protokolů. Pomocí parametru *Display Filter* si lze mezi zachycenými protokoly filtrovat požadovaný protokol. Vztažné veličiny grafů si lze upravit pomocí parametrů *Y Axis* a *Y Field*. Ukázka tvorby grafu programem Wireshark je na obrázku A.7.



Obr. A.7: Ukázka tvorby grafů v programu Wireshark.

Další typ možného zobrazení zachycené komunikace je pomocí nástroje **Flow graph**, který dokáže zachycenou komunikaci zobrazit v přehledném vývojovém diagramu.

A.4 Samostatné úkoly

1. Zachyťte síťovou komunikaci při stahování souborů z FTP a TFTP serveru na klientskou stanici.
2. V rámci analýzy zachycených dat se zaměřte na komunikaci transportních protokolů, které byly použity při stahování souborů z obou serverů. Jakým způsobem došlo k navázání spojení jednotlivých protokolů?
3. Dle zachycené komunikace vyobrazte proces navázaní spojení protokolu TCP s reálnými hodnotami pořadových čísel pomoci funkce Flow graph.
4. Pomocí programu Wireshark a funkce I/O Graph vygenerujte graf porovnávající maximální hodnoty přenesených bajtů protokolů TCP a UDP při stahování souboru *picture.jpg* ze serveru FTP a TFTP. Pro zřetelnost se pokuste provést jednotlivá stažení ze serverů co nejrychleji po sobě.
5. Omezte parametry přenosu pomocí prvku NETem. Nastavte šířku pásma na 50 Kb/s a ztrátovost paketů nastavte na 30%. Spusťte nové zachytávání paketů pomocí programu Wireshark a vyzkoušejte nové stažení testovacího souboru z obou serverů FTP a TFTP. Z nově zachycených dat proveďte analýzu chování transportních protokolů TCP a UDP při změně přenosových parametrů.
6. Získané poznatky shrňte do přehledné zprávy o měření.
7. Do zprávy o měření odpovězte také na doplňující otázky.

Doplňující otázky

- Jaký transportní protokol a port používá zachycený protokol FTP?
- Jaký transportní protokol a port používá zachycený protokol TFTP?
- Jaké příznaky sebou nese TCP segment, který je označen jako TCP Retransmission?

B Laboratorní úloha: Skupinové vysílání multicast

B.1 Zadání

- Seznamte se se simulačním prostředím GNS3.
- V simulačním prostředí GNS3 dle zadaného schématu nakonfigurujte síťové prvky tak, aby byla umožněna síťová komunikace všech zadaných prvků.
- Zprovozněte živé vysílání a následné zachycení video pomocí unicast a multicast vysílání.
- Pomocí programu Wireshark analyzujete síťovou komunikaci v rámci vytvořené sítě při vysílání typu unicast a při skupinovém vysílání pomocí protokolu PIM ve variantách Sparse a Dense Mode. Zaměřte se především na průběh komunikace protokolů IGMP a PIM také na výslednou trasu, přes kterou jednotlivé komunikace prochází.
- Dle zachycené komunikace definujte rozdíly mezi vysíláním unicast a multicast.
- Definujte hlavní rozdíly mezi variantami Sparse a Dense Mode skupinového směrovacího protokolu PIM.

B.2 Teoretický úvod

B.2.1 Základní způsoby síťové komunikace:

- **Unicast** - způsob komunikace, kdy jsou data přenášena z jednoho konkrétního prvku sítě na druhý. Při unicastovém přenosu je vždy jeden vysílač a jeden přijímač.
- **Multicast** - vícesměrové nebo také tzv. skupinové vysílání. Pomocí multicast komunikace je ke zdroji vysílání připojena vybraná skupina příjemců, kterým jsou data zasílána. V rámci IPv4 sítí je pro skupinové vysílání definována oblast rozsahu dostupných IP adres 224.0.0.0 – 239.255.255.255. Tento rozsah je definován jako vyhrazený rozsah třídy D.
- **Broadcast** - všesměrové vysílání, které umožňuje, aby jeden prvek v sítí odeslal data na všechny ostatní aktivní prvky v dané síti.

B.2.2 Protokol IGMP

Protokol IGMP (Internet Group Management Protocol) lze považovat jako doplněk protokolu IPv4, který umožňuje použití skupinového (vícesměrového) vysílání

v rámci IPv4 sítě. Hlavní úlohou uvedeného protokolu je správa multicastových skupin. Koncové stanice, tak pomocí IGMP protokolu, mohou požádat o přístup do skupinové adresy svůj přidružený směrovač. Tento směrovač je pak na základě IGMP zpráv informován, kterému zařízení přísluší adresa přijatého skupinové vysílání. Protokol IGMP existuje ve 3 variantách a to konkrétně IGMPv1, IGMPv2 a IGMPv3.

IGMPv1

Základní verze protokolu, jež má k dispozici pouze dva typy zpráv:

- Membership Query - zpráva, pomocí které zjišťuje směrovač od lokálních stanic jejich žádosti o členství skupinových adres. Tato zpráva je směrovači zasílána periodicky (výchozí hodnota je nastavena na 60 vteřin) na skupinovou adresu 224.0.0.1.
- Membership Report - odpověď na předchozí zprávu, kde lokální stanice posílají adresu skupinového vysílání, ke které se chtějí připojit. Aby nedocházelo k zahlcení v případě odpovědí většího počtu lokálních stanic, bývá tato zpráva odesílána v náhodně zvoleném časovém intervalu.

Pokud stanice během určitého časového intervalu neinformuje zprávou Membership Report o své žádosti pro danou skupinovou adresu, dochází k zastavení skupinového vysílání pro tuto stanici.

IGMPv2

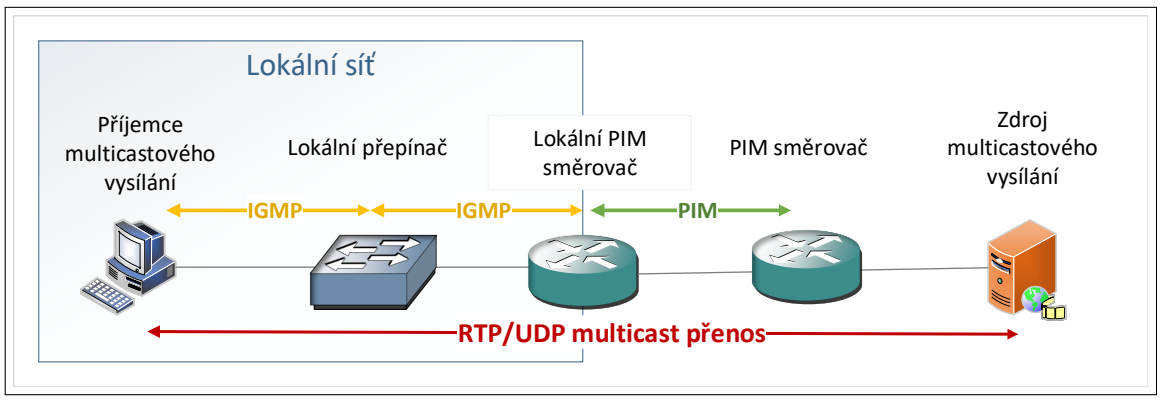
Druhá verze protokolu IGMP, obsahující 4 typy zpráv:

- Membership Query - podobná zpráva jako u předchozí verze, zasílaná na adresu 224.0.0.1 periodicky. Tentokrát je výchozí hodnota nastavena na 125 vteřin. Existují dva podtypy této zprávy:
  - General - dotaz na všechny skupiny, ke kterým jsou připojené lokální stanice.
  - Group-specific - dotaz zda existuje zájemce (odběratel) pro konkrétní specifikovanou skupinu.
- Membership Report verze 2 - odpověď na dotaz protokolu IGMPv2, na kterou odpovídá pouze jeden člen skupiny.
- Membership Report verze 1 - zpětně kompatibilní odpověď na dotaz protokolu IGMPv1.
- Leave Group - zpráva zasílána na adresu 224.0.0.2, pomocí které lokální stanice informuje o ukončení vysílaní a tím i o zrušení členství pro vybranou skupinovou adresu.

IGMPv2 vylepšuje předchozí protokol především v efektivitě správy skupin multicastového vysílání. V případě většího počtu směrovačů posílá na rozdíl od předchozí verze zprávu Membership Query pouze jeden směrovač. Další vylepšení spočívá v tom, že lokální stanice nově umí poslat zprávu, jež informuje o ukončení vysílání pro danou skupinu. Upravena je také hlavička protokolu, kde se nově vyskytuje pole určující časový limit, během kterého musí lokální stanice odpovědět na výzvu Membership Query.

B.2.3 PIM

Protokol PIM (Protocol Independent Multicast) patří do skupiny protokolů, které mají na starosti směrování multicastového přenosu mezi směrovači. V součinnosti s protokolem IGMP je možné pomocí protokolu PIM v IPv4 sítí sestavit a provozovat kompletní vícesměrové vysílání. Ukázka zmíněné součinnosti protokolu PIM a IGMP je zobrazeno na obrázku B.1.



Obr. B.1: Součinnost protokolu PIM a IGMP v IPv4 síti.

Přestože je protokol PIM považován jako protokol směrovací, tak si ve skutečnosti nevede vlastní směrovací tabulku. Směrovací informace odebírá od libovolného směrovacího protokolu unicastového přenosu. Protokol PIM, tak může spolupracovat s protokoly OSPF (Open Shortest Path First), RIP (Routing Information Protocol) a podobně. Protokol PIM lze použít v různých variantách. Na základě zvolené varianty protokolu PIM jsou sestavovány distribuční stromy různými způsoby. Základní varianty protokolu PIM jsou:

- Dense Mode (DM),
- Sparse Mode (SM),
- Sparse-Dense Mode (SDM),
- Rozšiřující varianty:
  - Bidirectional PIM (BIDIR-PIM),
  - Source-specific multicast (SSM).

Dense Mode (DM)

Varianta směrování protokolu, která je výhodná zejména u takových sítí, kde se předpokládá, že téměř všichni účastníci budou přijímat skupinové vysílání. Mechanismus varianty Dense Mode na začátku vysílání rozesílá (zaplavuje) provoz mezi všemi definovanými PIM směrovači. Pokud se stane, že určitý směrovač nemá připojené žádné zájemce o vysílání, tak tento směrovač odesílá zprávu pro potlačení daného skupinového vysílání. Potlačující zpráva (prune) má platnost 180 vteřin a po vypršení tohoto intervalu je znovu odesílána v případě, že daný směrovač stále nemá zájem o skupinové vysíláni. Naopak směrovače, které mají platné zájemce o připojení, pomocí mechanismu RPF (Reverse Path Forwarding) určují optimální trasu vysílání (strom nejkratších cest) a ostatní pakety, jež přichází mimo optimální trasu zahazují.

Sparse Mode (SM)

Na rozdíl od předchozí varianty se vytváří tzv. sdílené stromy za pomoci speciálně definovaného směrovače, označovaného jako RP (Rendezvous Point). Tento směrovač se stává kořenem sdíleného stromu směrovačů, přes který je řízeno veškeré skupinové vysílání. Ostatní směrovače, které jsou vybrány jako směrovače multicastového vysílání pro jednotlivé lokální sítě jsou označovány jako DR (Designed Router).

Princip Sparse Mode je koncipován tak, aby bylo skupinové vysílání zasíláno primárně pouze zájemcům o vysílání dané skupiny. Zájemci o skupinové vysílání se hlásí o daný příjem u RP opakovanými zprávami Join skrze routery DR. V případě, že již zájem nemají, informují o opuštění skupiny zprávou Prune. Ve fázi, kdy RP přijme žádosti příjemců, jsou vysílána data od zdroje vysílání nejdříve zapouzdřena a poté od svého přidruženého DR směrovače (pomocí zpráv Register) posílána na unicastovou adresu RP. RP data rozbalí a skrze vytvořený distribuční sdílený strom zasílá zvolenou cestou přes příslušné DR na přihlášené zájemce. Po ukončení registrace zdroje (Register-Stop zpráva) jsou již od zdroje dat přeposílána nativní data přes RP přímo ke příjemci. Varianta Sparse Mode však za určitých podmínek (například když dojde k vytížení RP směrovače) umožňuje posílat data přímo od zdroje nejkratší cestou přímo k cílovému příjemci.

Sparse-Dense Mode (SDM)

Kombinace obou předchozích variant, jež je používána například na zařízeních Cisco. Pokud není žádný směrovač v sítí definován jako RP, funguje skupinové směrování na principu Dense Mode. V opačném případě je směrování řízeno dle principu Sparse Mode.

Typy zpráv protokolu PIM

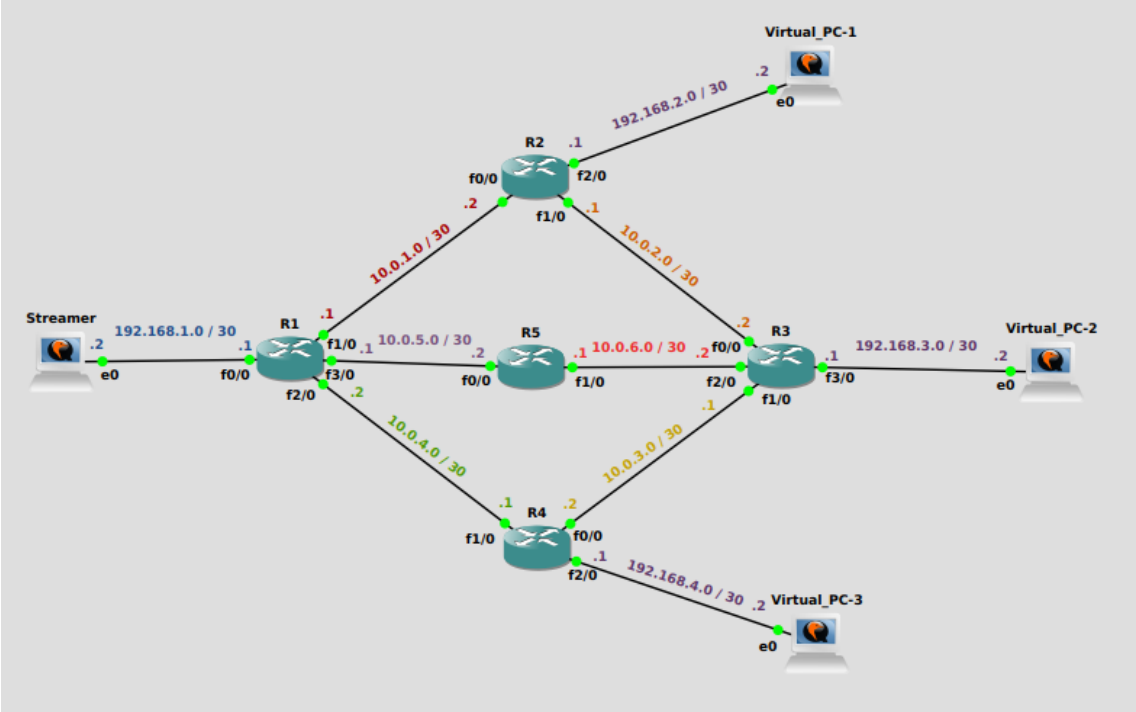
Protokol PIM ke svému korektnímu fungování používá několik typu zpráv, jež jsou většinou zasílány na adresu 224.0.0.13, která definuje všechny směrovače, kde je nakonfigurovaný protokol PIM. Základní zprávy protokolu PIM jsou:

- Hello - slouží k detekci všech směrovačů, kde je definován a nastaven protokol PIM. Zasílaná periodicky, výchozí interval opakování je 30 vteřin.
- Register (pouze u SM) - zprávu odesílá PIM směrovač, ke kterému je připojený zdroj vysílání. Cílem je unicastová adresa RP směrovače, který je touto zprávou (zapouzdřenou do unicastového vysílání) informován o aktivním vysíláním zdroje multicastových dat.
- Register-Stop (pouze u SM) - reakce RP směrovače na přijatou zprávu Register, kterou je DR směrovač zdroje informován o úspěšné registraci. Tímto DR směrovač zdroje ukončí posílání zapouzdřených dat a vytvořeným zdrojovým stromem posílá nezapouzdřená nativní data k RP, který je poté dál rozesílá k příjemci .
- Join/Prune - zpráva která ve svém těle, obsahuje seznamy zdrojů multicastového vysílání, u kterých dané směrovače chtějí pokračovat v připojení (Join), nebo o jejich vysílání již nemají zájem (Prune).
- Bootstrap (pouze u SM) - využívá mechanismy automatického delegování RP směrovače v rámci dané sítě.
- Assert - detekuje a zabraňuje duplicitě vysílaných tras. Děje se tak především u varianty DM, kdy na směrovač, který žádá o přijetí do skupiny a vysílání k němu dorazí ze 2 či více směru. Cílový směrovač si poté vybere vítězný směrovač, jehož zvolí na základě porovnání parametrů použitého unicastového směrovacího protokolu (Metric Preference), metriky a IP adresy.
- Graft (pouze u DM) - podobná zprávě Join s tím rozdílem, že tato zpráva slouží k opětovnému navázání spojení, které bylo v minulosti již navázáno a později přerušeno. Využívá se pouze ve variantě Dense Mode a zasílána je unicastovou adresu směrovačů.
- Graft-Ack (pouze u DM) - zpráva, která potvrzuje přijetí zprávy Graft, jež má využití pouze u varianty Dense Mode.
- Candidate-RP-Advertisement (pouze u SM) - směrovač touto zprávou ohlašuje zájem o kandidaturu na RP při mechanizmu automatického volení RP směrovače.

B.3 Vypracování

B.3.1 Příprava topologie v testovacím prostředí GNS3

Po spuštění virtualizace GNS3_LAB pomocí programu VMware spusťte simulační prostředí GNS3 a vytvořte nový projekt, který si libovolně pojmenujte. Heslo k uživatelskému účtu GNS3_LAB pro OS Ubuntu je: **student**. Dle níže uvedeného návodu nakonfigurujte a zprovozněte zapojení všech síťových prvků dle obrázku B.2.



Obr. B.2: Schéma zapojení pro demonstraci skupinového vysílání.

Konfigurace routerů Cisco 3600

Z nabídky všech předinstalovaných komponent (*Browse all device*) vyberte zařízení s názvem **C3600** a přetáhněte jej do pracovní plochy GNS3. Dle zadaného schématu je nutné do pracovní plochy přesunout všech 5 zařízení a poté je mezi sebou propojit pomocí nástroje **Add a link**. Pro přehled jednotlivých síťových rozhraní směrovače a pozdější korektní konfiguraci se ujistěte, že je aktivovaná funkce zobrazení popisků síťových rozhraní. Toto nastavení je možné aktivovat na hlavním panelu aplikace GNS3 v záložce *View - Show/Hide interface label*. Instance C3600 je virtuální stanice simulující reálný směrovač Cisco 3600 pracující operačním systému. Pro konfiguraci směrovače se připojte k uživatelskému textovému rozhraní pomocí

připojení Telnet. Připojení k uživatelskému rozhraní IOS je možné provést kliknutím pravého tlačítka myši a zvolením možnosti *Console*. Před připojením k danému síťového prvku je nutné mít požadovaný prvek spuštěný. To lze provést vybráním možnosti *Start* u konkrétního prvku, nebo kliknutím na *Start/resume all nodes* na hlavním panelu, při kterém se spustí všechny prvky na pracovní ploše. Po úspěšném připojení k rozhraní směrovače se otevře nové okno s příkazovou řádkou. Nyní je možné postupně vkládat textové příkazy.

Pro umožnění konfigurace routeru je nutné přejít konfiguračního módu zadáním příkazu:

```
R1#conf t
```

Pro správnou konfiguraci skupinového vysílání je nutné na všech směrovačích povolit skupinové vysílání příkazem:

```
R1(config)#ip multicast-routing
```

Dále je nutné u všech směrovačů provést konfiguraci všech aktivních síťových rozhraní a to nejdříve nastavením odpovídající IPv4 adresy a síťové masky příkazy:

```
R1(config)#int f0/0
R1(config-if)#ip add 192.168.1.1 255.255.255.252
```

Pro směrování skupinového vysílání mezi směrovači bude využito protokolu PIM. Na všech aktivních síťových rozhraní je tedy nutné definovat tento protokol a vybranou variantu protokolu PIM. V rámci laboratorní úlohy je zvolena varianta SDM (Sparse Dense Mode), která poskytuje výhodu v tom, že je možné operativně měnit varianty skupinového směrování mezi variantou Sparse a Dense jen podle toho, zda je definován RP (Rendezvous Point). Pokud není žádný směrovač v sítí definován jako RP, funguje skupinové směrování na principu Dense Mode. V opačném případě je směrování řízeno na principu Sparse Mode. Konfigurace protokolu PIM ve zvolené variantě je provedena příkazem:

```
R1(config-if)#ip pim sparse-dense-mode
```

Na síťových rozhraních směrovačů, ke kterým jsou připojené klientské stanice, které budou přijímat nebo vysílat multicastový přenos (instance Streamer, Virtual_PC-2 a Virtual_PC-3), je nutné definovat zvolenou skupinovou adresu příkazem:

```
R1(config-if)#ip igmp join-group 224.24.24.24
```

Při konfiguraci síťových rozhraní je také nutné dané rozhraní aktivovat a následně zapsat konfigurační nastavení. K tomu slouží příkazy:

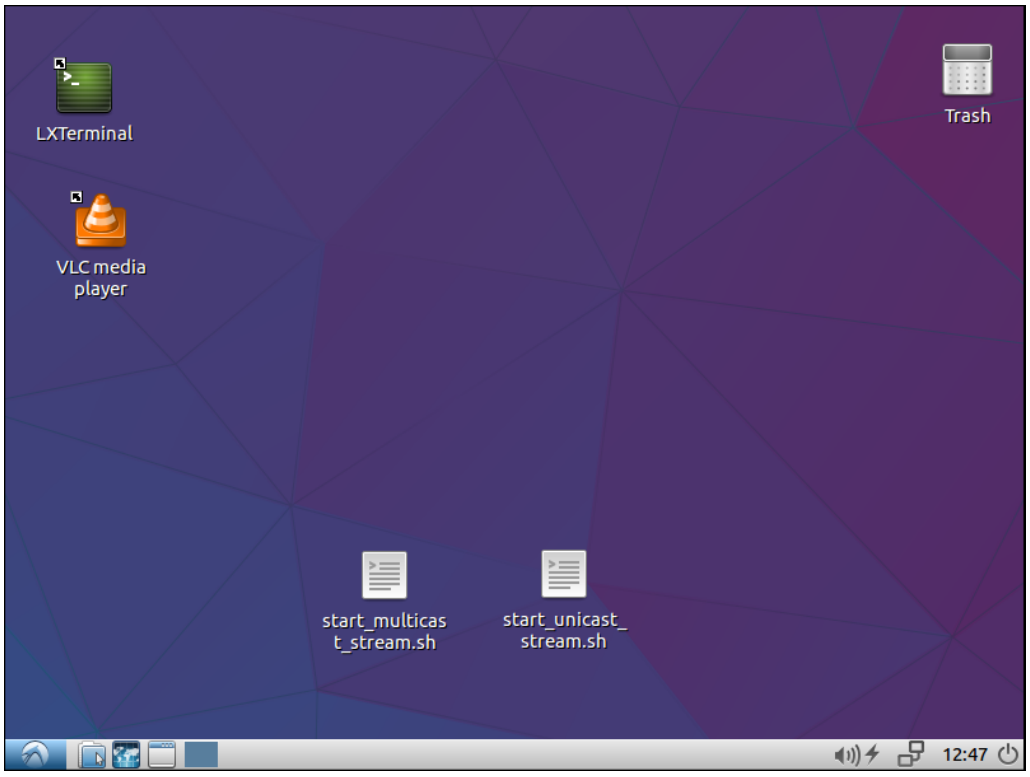
```
R1(config-if)#no shut
R1(config-if)#do wr
```

Aby bylo možné v rámci laboratorní úlohy demonstrovat rozdíly mezi unicast a multicast vysíláním, je nutné na všech směrovačích definovat i klasický směrovací protokol, který zaručí, že všechny síťové prvky úlohy budou mezi sebou dostupné. V tomto případě je použit směrovací protokol OSPF. Na jednotlivých směrovačích je poté nutné definovat všechny přidružené sítě. Definování sítí je provedeno IPv4 adresou sítě, příslušnou wildcard maskou a definováním čísla oblasti. Ukázka konfigurace protokolu OSPF na směrovači R1:

```
R1(config)#router ospf 1
R1(config-router)#network 192.168.1.0 0.0.0.3 area 0
R1(config-router)#network 10.0.1.0 0.0.0.3 area 0
```

Konfigurace klientských stanic

Klientská stanice bude sloužit k vysílání a zachycení přenosu videa v reálném čase přes vytvořenou síť směrovačů. K tomuto úkolu byla vytvořena instance s názvem Virtual_PC. Tato instance běží na zjednodušeném operačním systému Lubuntu 16.04.3. V případě požadovaného přihlášení je přihlašovací heslo stejné jako název uživatelského účtu: student. Náhled na grafické prostředí vytvořené instance Virtual_PC je na obrázku B.3.



Obr. B.3: Ukázka klientské stanice Virtual_PC.

K vysílání a zachycení videa použijte na instanci Virtual_PC předinstalovaného programu VLC media player. K usnadnění spouštění proudového vysílání přes program VLC, jež bude probíhat pouze na klientské stanici Streamer (viz. obrázek B.2), využijte dvou vytvořených skriptů, které naleznete na ploše instance Virtual_PC. Skript pro multicastové vysílání s názvem **start_multicast_stream.sh** po spuštění vysílá video s názvem multicast_stream.mp4 pomocí protokolu RTP (port 5004) na skupinovou IPv4 adresu 224.24.24.24. Druhý skript pojmenovaný názvem **start_unicast_stream.sh** má za úkol vysílat video s názvem unicast_stream.mp4 pomocí protokolu UDP (port 1234) na IP adresu instance Virtual_PC-1 192.168.2.2. Oba skripty je možné spustit zároveň nezávisle na sobě. Pro ukončení vysílání je nutné vypnout okno aplikace VLC s příslušnou vysílací adresou. Pro ověření správné konfigurace a tím i ověření, zda oba typy vysílání fungují správně zachyťte vysílané video na příslušných klientských stanicích. Po spuštění VLC playeru a kliknutím na položku *Media* vyberte položku *Open Network Stream...* Pro zachycení skupinového vysílání (na instancích Virtual_PC-2 a Virtual_PC-3) zadejte následující adresu vysílání:

```
rtp://@224.24.24.24:5004
```

Pro zachycení unicast vysílání (na instanci Virtual_PC-1) je nutné zadat adresu:

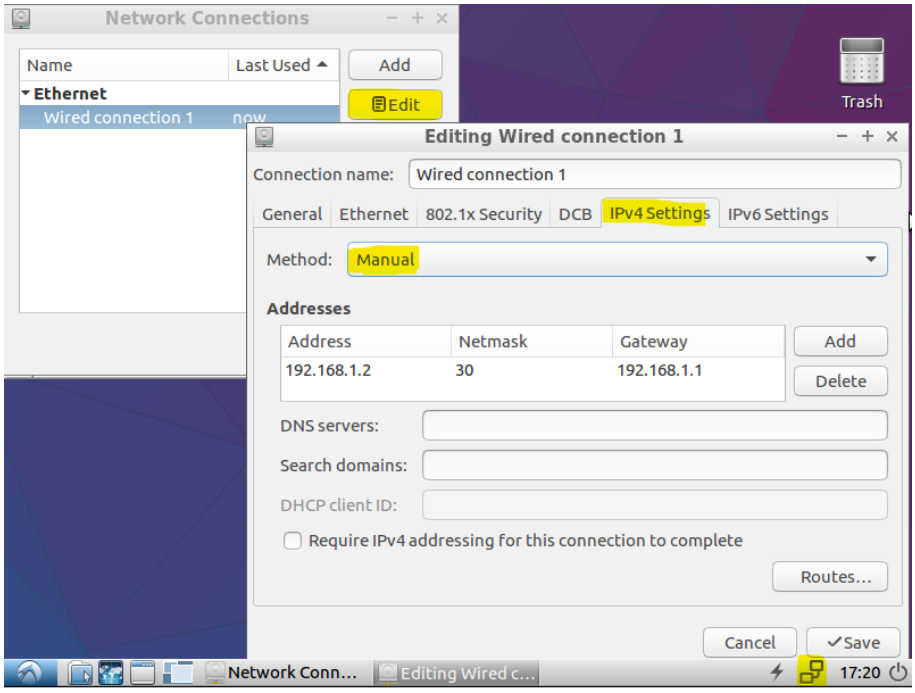
```
udp://@192.168.2.2:1234
```

Při správné konfiguraci se po stisknutí tlačítka *Play* spustí patřičné vysílání.

Předinstalovanou instanci **Virtual_PC** je možné najít v položce (*Browse all device*) a přesunutím do pracovní plochy lze vybranou instanci přidat do projektu. Po přidání všech 4 zařízení je nutné je dle zadaného schématu spojit umístit na vhodné místo a pomocí nástroje **Add a link** dané instance propojit s konkrétními směrovači. Umístěné instance je také možné volitelně pojmenovat kliknutím pravého tlačítka myši a zvolením položky *Rename*. Pro připojení na dané klientské stanice je nutné samotné instance spustit stisknutím pravého tlačítka myši a vybráním položky *Start*. Poté se je možné připojit přes do VNC rozhraní stisknutím pravého tlačítka myši a vybráním položky *Console*.

Po připojení na klientskou stanici je nutné správně nastavit patřičnou IP adresu, síťovou masku a výchozí bránu. Výchozí brána je vždy IP adresa směrovače, ke kterému je daná klientská stanice připojena. Síťové nastavení je možné konfigurovat přes grafické rozhraní kliknutím na ikonu síťového adaptéru (na spodní liště OS Lubuntu) a vybráním položky *Edit Connection....* Dále je nutné vybrat jediné Ethernetové spojení (ve výchozím nastavení pojmenovanou Wired connection 1) a kliknout na tlačítko *Edit*. Nyní se v záložce Ethernet ujistěte, že je vybrána volba Multicast a pokračujte na záložku IPv4 settings. Zde vyberte manuální nastavení

a příslušné síťovou konfiguraci nastavte. Nezapomeňte své nastavení uložit kliknutím na tlačítko *Save*. Ukázka správné konfigurace klientské stanice (Streamer) se nachází na obrázku B.4.



Obr. B.4: Ukázka síťového nastavení na instancích klientských stanic.

Pro zavedení změn v síťovém nastavení je ještě nutné dané připojení (Wired connection 1) deaktivovat a znovu aktivovat. Učinit tak můžete kliknutím na ikonu síťového adaptéru, vybraním volby *Disconnect* a poté opět aktivovat kliknutím na položku *Wired connection 1*. Správnost síťového nastavení je možné ověřit výpisem příkazu `ifconfig` při spuštění terminálového nástroje LXTerminal, případně pomocí příkazu `ping` na již nakonfigurované síťové prvky v rámci projektu.

Ukázka správné konfigurace směrovače R1

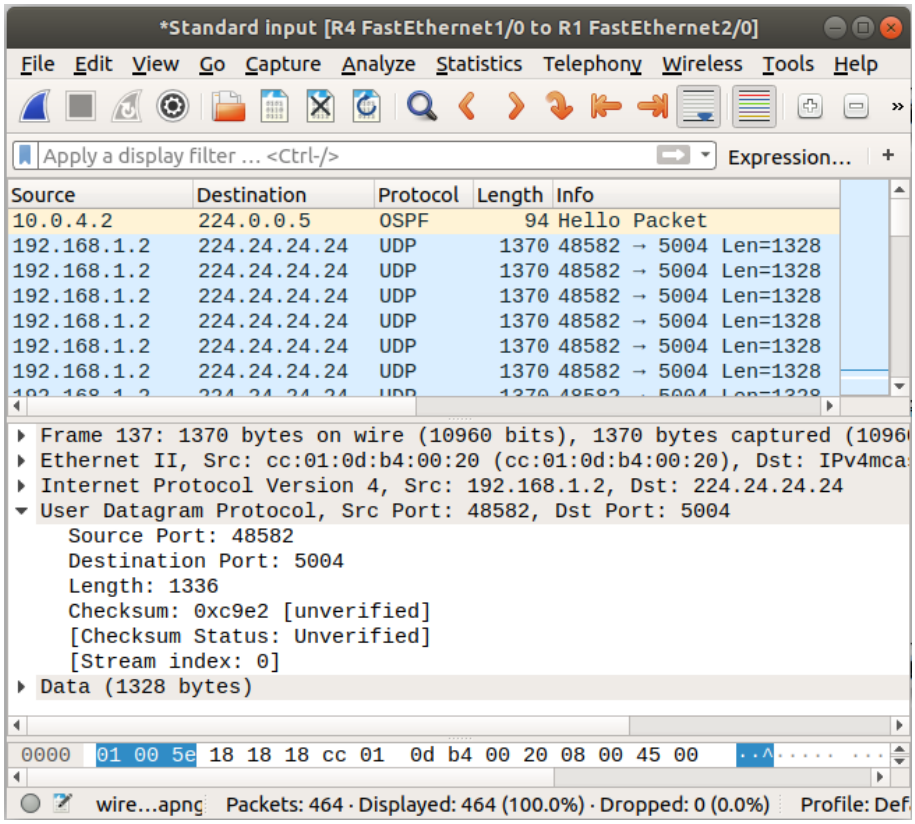
```
R1#conf t
R1(config)#ip multicast-routing
R1(config)#int f0/0
R1(config-if)#ip add 192.168.1.1 255.255.255.252
R1(config-if)#ip pim sparse-dense-mode
R1(config-if)#ip igmp join-group 224.24.24.24
R1(config-if)#no shut

R1(config-if)#int f1/0
```

```
R1(config-if)#ip add 10.0.1.1 255.255.255.252
R1(config-if)#ip pim sparse-dense-mode
R1(config-if)#no shut

R1(config)#int f3/0
R1(config-if)#ip add 10.0.5.1 255.255.255.252
R1(config-if)#ip pim sparse-dense-mode
R1(config-if)#no shut

R1(config-if)#int f2/0
R1(config-if)#ip add 10.0.4.2 255.255.255.252
R1(config-if)#ip pim sparse-dense-mode
R1(config-if)#no shut
R1(config-if)#do wr

R1(config-if)#exit
R1(config)#router ospf 1
R1(config-router)#network 192.168.1.0 0.0.0.3 area 0
R1(config-router)#network 10.0.1.0 0.0.0.3 area 0
R1(config-router)#network 10.0.5.0 0.0.0.3 area 0
R1(config-router)#network 10.0.4.0 0.0.0.3 area 0
R1(config-router)#do wr
```

B.3.2 Zachycení síťového provozu

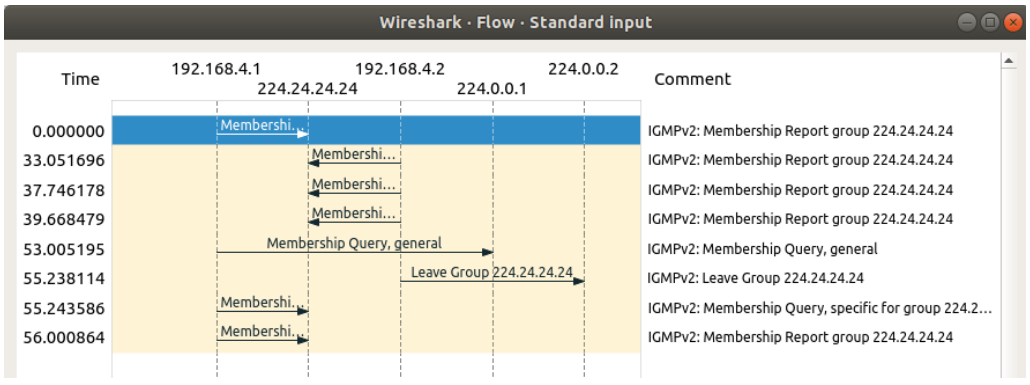
Pro zachycení síťového provozu použijte program Wireshark, který je již předinstalovaný v simulačním prostředí GNS3. Zachytávání paketů pomocí Wiresharku je možné spustit kliknutím pravého tlačítka myši na požadovaný síťový propoj a vybráním volby *Start capture*.

Před automatickým spuštěním programu Wireshark je nutné určit název souboru, do kterého se budou zachycená data ukládat. Aktivní zachytávání síťové komunikace indikuje obrázek lupy na síťovém spojení. Pokud se aplikace Wireshark nespustí automaticky, je možné ji manuálně spustit výběrem možnosti *Start Wireshark* po stisku pravého tlačítka myši na vybraném spoji. Pro ukončení daného zachycení stiskněte na hlavním panelu programu Wireshark tlačítko *Stop capturing packets* a soubor se zachycenými daty si uložte ve formátu .pcapng pro pozdější analýzu. Pro další měření je nutné ukončit aplikaci Wireshark a také zastavit zachytávání paketů v GNS3 kliknutím pravého tlačítka myši na ikonu lupy (analyzovaný spoj) a vybraním možnosti *Stop capture*. Ukázka správného zachycení paketů je na obrázku B.5.



Obr. B.5: Ukázka správného průběhu skupinového vysílání.

Jednou z používaných funkcí programu Wireshark je tvorba vývojového diagramu zachycené komunikace **Flow graph**, kterou najdete pod záložkou Statistics na hlavní liště. Pomocí parametru *Display Filter* si lze mezi zachycenými protokoly filtrovat právě vybraný protokol. Ukázka tvorby grafu programem Wireshark je na obrázku B.6.



Obr. B.6: Ukázka tvorby grafů v programu Wireshark.

B.4 Samostatné úkoly

1. Dle návodu nakonfigurujte všechny zadané síťové prvky tak, aby byla zajištěna síťová komunikace mezi všemi zadanými prvky. Dostupnost všech prvků ověřte pomocí nástroje PING a TRACEROUTE v aplikaci Terminal.
2. Na instanci Streamer spusťte skripty generující multicast a unicast vysílání. Správnost konfigurace ověřte zachycením a spuštěním vysílaného videa na instanci Virtual_PC-1 pro unicast vysílání a pro vysílání multicast na instancích Virtual_PC-2 a Virtual_PC-3.
3. Pomocí programu Wireshark proveďte analýzu probíhající síťové komunikace mezi všemi směrovači při spouštění jednotlivých skriptů, jež generují jednotlivá vysílání. Dle analýzy komunikace se zaměřte na trasu unicastového a multicastového vysílání, které je v tomto případě zajištěno pomocí protokolu PIM ve variantě Dense Mode. Analyzujte zachycené zprávy protokol PIM a zjištěnou trasu komunikace Unicast a Multicast (PIM Dense Mode) zaznamenejte do obrázku B.7, který naleznete v příloze. V případě PIM protokolu se zaměřte především na chování protokolu při spuštění multicastového vysílání.
4. Dále se zaměřte na multicast komunikaci mezi směrovačem R3 a instancí Virtual_PC-2. Při spuštěném zachytávání síťové komunikace pomocí programu Wireshark ukončete a znovu spusťte multicastové vysílání. Pomocí funkce Flow Graf (v aplikaci Wireshark) zobrazte vývojový graf protokolu IGMP mezi směrovačem R4 a instancí Virtual_PC-3 při navázání, průběhu a při ukončení skupinového vysílání.
5. Proveďte změnu typu směrování protokolu PIM z Sparse na Dense Mode. Změnu proveďte definováním RP směrovače, kterým bude směrovač R5, definovaný IP adresou 10.0.5.1. Na všech směrovačích je nutné zadat příkaz:

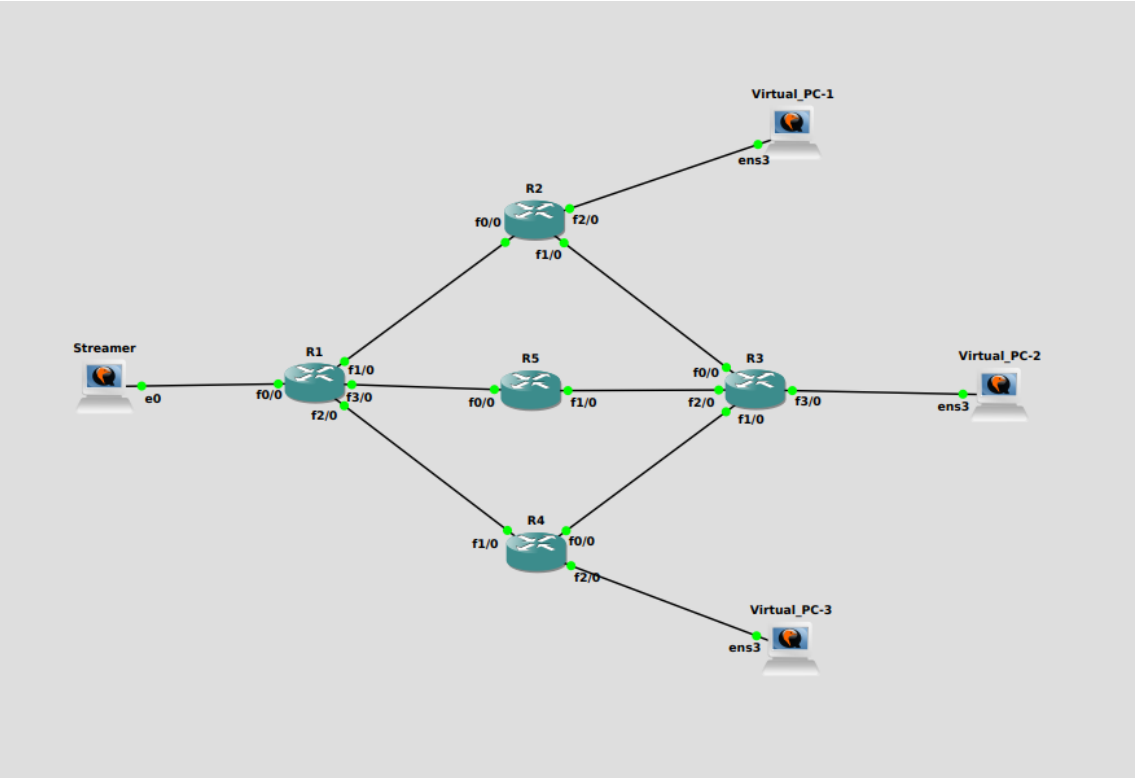
   ```
   R1(config-if)#ip pim rp-address 10.0.5.2
   ```

6. V případě PIM Sparse Mode se nyní zaměřte na komunikaci mezi směrovači R1-R4. Při zapnutém zachytávání síťové komunikace spusťte vysílání multicast a vysílání zobrazte na stanicích Virtual_PC-2 a Virtual_PC-3. Pomocí funkce Flow Graf vygenerujte vývojový graf PIM protokolu ve variantě Sparse Mode.
7. Obdobně jako v bodu č. 3 proveďte analýzu cesty zahájení komunikace při směrování protokolu PIM pomocí nově nastavené varianty Sparse Mode. Novou cestu zaznamenejte do obrázku B.7
8. Získané poznatky shrňte do přehledné zprávy o měření.
9. Do zprávy o měření odpovězte také na doplňující otázky.

Doplňující otázky

- Jaké jsou hlavní rozdíly mezi unicast a multicast vysíláním?
- Který protokol a které aplikační port je použit pro skupinové vysílání pomocí aplikace VLC?
- Uveďte typy zpráv protokolu IGMP, které se podařilo zachytit?
- Jakou podobu má MAC adresa skupinových IPv4 adres?
- Jakou cílovou IPv4 adresu používá protokol IGMP při odeslání zprávy typu Leave Group?
- Jaké jsou hlavní rozdíly mezi variantami protokolu PIM Sparse a Dense Mode?
- Jaká je hlavní úloha směrovače označeného jak RP (Rendezvous Point)?
- Jaké zprávy protokolu PIM se používají mezi směrovači ve variantě Sparse Mode?
- Jaké zprávy protokolu PIM se používají mezi směrovači ve variantě Dense Mode?

B.5 Příloha



Obr. B.7: Šablona pro zakreslení cest požadovaných vysílání.

C Laboratorní úloha: Základy penetračního testování

C.1 Zadání

- Seznamte se se simulačním prostředím GNS3.
- V rámci projektu GNS3 připojte instanci Kali Linux k testované síti.
- Nalezněte IP adresy aktivních síťových zařízení v rámci testované sítě.
- Pomocí programu Nmap detekujte na nalezených IP adresách otevřené TCP a UDP porty a příslušné běžící aplikační služby.
- Generujte DoS útok, pomocí kterého bude odstavena služba HTTP stránky na detekovaném WWW serveru.
- Proveďte prolomení přístupu na nalezený FTP server.

C.2 Teoretický úvod

Penetrační testování

Penetrační testování slouží k odhalování možných zranitelností síťových zařízení nebo sítí jako takových. Ekvivalentním pojmenováním pro pojem penetrační testování se udává také jako etický hacking, který princip testovaní popisuje lépe. Výsledkem penetračního testování většinou bývá takzvaný bezpečnostní audit, který správcům testované infrastruktury může podat důležité informace o kvalitě zabezpečení svého systému. Zranitelnosti se tak velmi často projeví ve špatné konfiguraci jednotlivých prvků systému, nebo v chybějících zabezpečení. Penetrační testy se mohou z pohledu způsobu provedení a cíle rozdělit do několika typů. Nejčastější typy penetračního testování jsou:

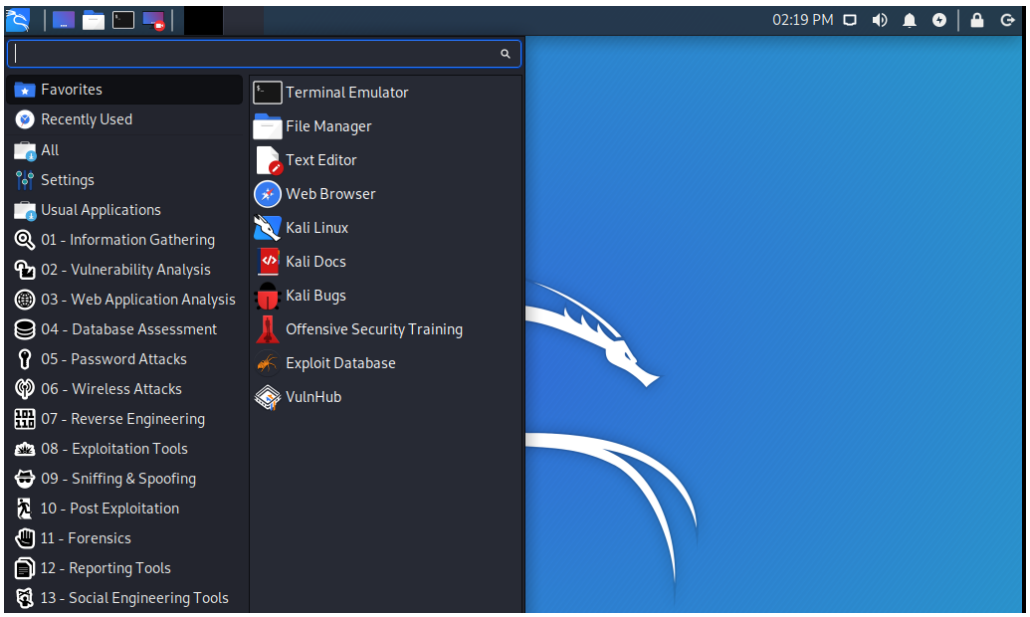
- Externí penetrační testy - simulace potenciálních útoku z vnější strany testované sítě.
- Zátěžové penetrační testy - simulace útoků DoS (DDoS), jež mají za následek odstavení poskytovaných služeb zátěžovým přetížením serverů.
- Interní penetrační testy - testování zranitelností, které mohou způsobit samotní uživatelé daných systémů.

Kali Linux

Kali Linux je volně dostupná linuxová distribuce, která obsahuje spoustu nástrojů sloužících k vykonání bezpečnostních analýz sítí a tzv. penetračnímu testování. Za zmínku stojí nejznámější nástroje:

- Nmap,
- Metasploit Framework,
- Hydra,
- Aircrack–ng.

Za vývojem distribuce Kali Linux stojí komunita IT specialistů, kteří na svém webu [1] poskytují veškeré informace včetně dokumentace produktu, aktivního fóra i tutoriálů zaměřující se na práci s Kali Linux. V rámci laboratorní úlohy bude použita nejnovější verze z března roku 2020 označená jako Kali Linux 2020.1a. Ukázka prostředí virtualizace Kali Linux je na obrázku C.1



Obr. C.1: Ukázka prostředí Kali Linux.

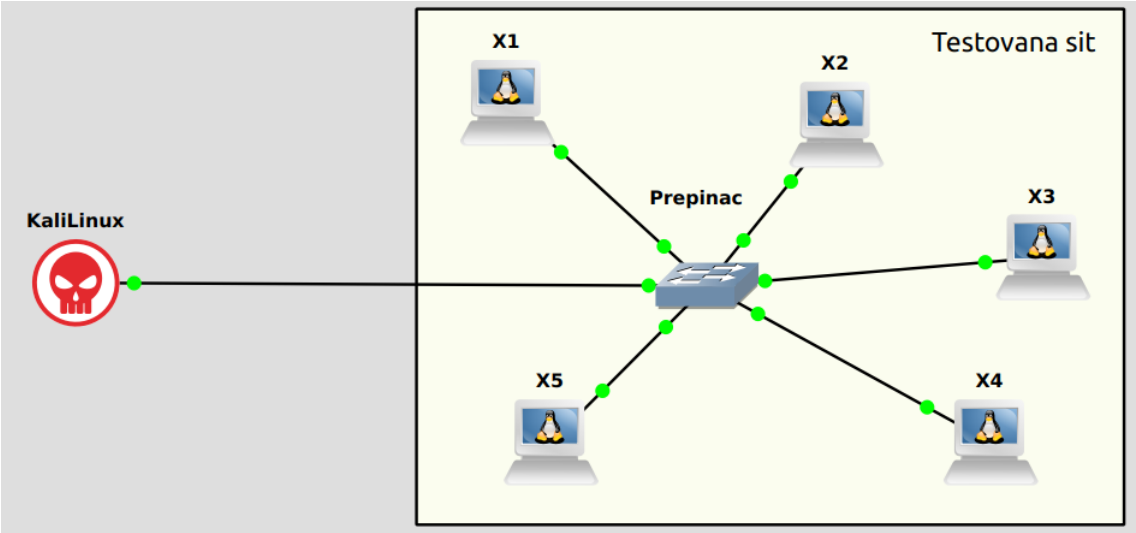
C.3 Vypracování

C.3.1 Příprava topologie v testovacím prostředí GNS3

Po spuštění virtualizace GNS3_LAB3 pomocí programu VMware spusťte simulační prostředí GNS3 a otevřete již vytvořený projekt PenTest z položky Recent project. Heslo k uživatelskému účtu GNS3_LAB pro OS Ubuntu je: **student**.

Z nabídky všech předinstalovaných komponent (*Browse all device*) vyberte zařízení s názvem **Kali Linux** a přetáhněte jej do pracovní plochy GNS3. Poté jej pomocí nástroje **Add a link** propojte s volným síťovým portem na přípínači, viz schéma zapojení na obrázku C.2. Pro připojení na instanci Kali Linux je nutné

[1] https://www.kali.org



Obr. C.2: Schéma zapojení pro demonstraci základů penetračního testování.

instanci spustit stisknutím pravého tlačítka myši a vybráním položky *Start*. Dále je nutné spuštění všech předem připravených prvků projektu. To můžete provést kliknutím na tlačítko **Start/Resume all nodes** na hlavní panelu aplikace GNS3. Poté je možné se připojit přes VNC rozhraní instance KaliLinux stisknutím pravého tlačítka myši na danou instanci a vybráním položky *Console*. Přihlašovací údaje instance Kali Linux jsou:

- Přihlašovací jméno: kali
- Heslo: student

Po úspěšném přihlášení se instanci přiřadí IP adresa z DHCP serveru testované sítě a je možné začít se zadanými úkoly dle níže uvedeného postupu.

C.3.2 Detekce aktivních síťových zařízení

Předtím než bude spuštěno vyhledávání, spusťte sledování síťového provozu pomocí programu Wireshark. Zachytávání paketů pomocí Wiresharku je možné spustit kliknutím pravého tlačítka myši na požadovaný síťový propoj a vybráním volby *Start capture*. V našem případě se bude jednat o propoj mezi instancí KaliLinux a přepínačem.

Před automatickým spuštěním programu Wireshark je nutné určit název souboru, do kterého se budou zachycená data ukládat. Aktivní zachytávání síťové komunikace indikuje obrázek lupy na síťovém spojení. Pokud se aplikace Wireshark nespustí automaticky, je možné ji manuálně spustit výběrem možnosti *Start Wireshark* po stisku pravého tlačítka myši na vybraném spoji. Pro ukončení daného zachycení stiskněte na hlavním panelu programu Wireshark tlačítko *Stop capturing*

packets. Soubor se zachycenými daty si je možné uložit ve formátu .pcapng pro pozdější analýzu. Pro další zachytávání síťové komunikace je nutné ukončit aplikaci Wireshark a také zastavit zachytávání paketů v GNS3 kliknutím pravého tlačítka myši na ikonu lupy (analyzovaný spoj) a vybraním možnosti *Stop capture*.

Pro zjištění aktivních síťových prvků bude využito nástroje **Netdiscover**. V tomto případě využijeme informace ze získané IP adresy od skrytého DHCP serveru. Aby bylo vyhledávání úspěšné zvolte IP adresu rozsahu s prefixem /16. Spuštění vyhledávání pomocí programu Netdiscover proveďte zadáním následujícího příkazu v terminálovém okně:

```
sudo netdiscover -r 192.168.0.0/16
```

Zjištěné IP adresy vyplňte do tabulky C.3.2 s příslušnými adresami MAC. Jednotlivé IP adresy zadávejte od nejnižší hodnoty poslední bajtu IP adresy po nejvyšší zjištěnou.

Tab. C.1: Tabuka pro vyplnění k analýze síťových prvků.

Server	IP adresa	MAC adresa	Otevřené TCP/UDP porty
X1			
X2			
X3			
X4			
X5			

Dále v programu Wireshark proveďte analýzu zachycených dat komunikace mezi instancí Kali Linux a testovanou sítí při spuštěném skenování. Odpovězte na první otázku v sekci C.3.6 Doplňující otázky.

C.3.3 Detekce otevřených TCP portů

Skenováním aktivních TCP portů získaných IP adres bude napovězeno, které síťové aplikace běží na objevených IP adresách síťových zařízení. Pro skenování síťových portů bude použit nástroj **Nmap**. V rámci tohoto úkolu použijte metodu TCP SYN skenování, která na všechny aplikační porty zvolené IP adresy zasílá TCP zprávy typu SYN a od každého aktivního aplikačního naslouchá korektní odpovědi TCP protokolu. Aplikaci Nmap je možné spustit z příkazové řádku aplikace Terminal a zmíněné SYN skenování je možné příkazem:

```
sudo nmap -sT <IP_Adresa>
```

Proveďte TCP - SYN skenování u všech zjištěných IP adres z předchozího úkolu a zjištěné otevřené TCP porty vložte do příslušného sloupce tabulky C.3.2. V programu Wireshark proveďte analýzu zachycených dat komunikace mezi instancí KaliLinux a testovanou sítí při spuštěném skenování jednotlivých IP adres. Zaměřte se na zachycené zprávy směřující na cílové porty, které se podařilo označit jako aktivní (open) [2]. Odpovězte na otázku 2 a 3 v sekci C.3.6 Doplňující otázky.

C.3.4 Detekce otevřených UDP portů

V tomto bodě bude vašim úkolem zjistit dostupné UDP porty na odhalených OP adresách. Metoda UDP skenování je však obtížnější a zdlouhavější v porovnání s předchozí metodou. Transportní protokol UDP neobsahuje metody třícestného navázání komunikace (protokol TCP), tudíž není možné ověřit dostupný port na základě příznaků, které UDP protokol ani neobsahuje. Metoda UDP skenování proto odesílá prázdné UDP pakety na aplikační porty dané IP adresy a naslouchá odpovědi protokolu ICMP. Pokud dorazí odpověď ICPM typ 3 (Port unreachable), tak je jasné, že je cílový port nedostupný. V případě UDP protokolu je k ověření otevřeného UDP portu nutné, aby se provedlo kompletní a korektní navázání spojení. Jelikož se často stane, že se kompletní komunikace nedokáže navázat, je nejčastější stav získané informace o UDP portu označován jako open/filtered. Stav open/filtered znamená, že nepřišla odpověď ICMP Port Unreachable, ale také se nepodařilo programu Nmap navázat korektní komunikaci, z důvodu například možného blokování firewallem.

Jak již bylo uvedeno, metoda skenování UDP je zdlouhavější a proto se zaměřte pouze na získání informací o otevřených nebo filtrovaných portech protokolu DNS, DHCP a TFTP). Skenování proveďte pomocí programu Nmap a to příkazem:

```
sudo nmap -sU -p 53,67,69 <IP_adresa>
```

Proveďte UDP skenování u všech objevených IP adres z prvního úkolu a zjištěné otevřené/filtrované UDP porty vložte opět do tabulky C.3.2. V programu Wireshark proveďte analýzu zachycených dat komunikace mezi instancí KaliLinux a testovanou sítí při spuštěném skenování jednotlivých IP adres. Zaměřte se na zachycené zprávy směřující na cílové porty, které se podařilo označit jako aktivní/filtrované (open/filtered). Odpovězte na otázku 4 a 5 v sekci C.3.6 Doplňující otázky.

[2] Program Wireshark umožňuje filtrovat komunikaci dle použitých aplikačních portů. Pro filtraci je možné v poli Filter zadat příkaz tcp.port==X (udp.port==X), který zobrazí pouze zprávy zadaného portu X.

C.3.5 DoS útok na HTTP server

V předchozích bodech se povedlo docílit získaní informací o dostupných síťových aplikací, které se vyskytují na daných IP adresách. Nyní se zaměříme na IP adresu (server), na kterém se podařilo nalézt aktivní port 80, jež značí, že na dané serveru poběží webový server. Ve webovém prohlížeči v instanci KaliLinux ověřte, že webový server na odhalené IP adrese opravdu běží otevřením webové stránky http://<IP_adresa>. Pokud se vám podařilo získat správnou IP adresu zobrazí se vám HTTP stránka s logem GNS3 a textem: Vítejte na testovací stránce.

Nyní proveďte DoS útok, který bude mít za úkol znepřístupnit tuto webovou stránku. Pro možnost generovat DoS útoky je možné využít volně dostupného softwaru s názvem **SlowHTTPTest**.

Program SlowHTTPtest využívá zranitelnosti nezabezpečených HTTP serverů. Program odesílá na cílenou adresu data velmi nízkou rychlostí a po velmi malých částech. Při takovém průběhu se u nezabezpečeného serveru zaplní kapacity přístupu na HTTP stránku a po tuto dobu je cílený server nedostupný pro další požadavky.

Program SlowHTTPtest lze ovládat pomocí příkazového řádku. Použijte následující příkaz:

```
sudo slowhttptest -c 1000 -H -i 10 -r 200 -t GET
-l 60 -g -o /home/kali/vystup_testu -u http://<IP_adresa>
```

Jednotlivé parametry znamenají následující:

- -c 1000 (určuje počet navázaných spojení),
- -H (typ útoky využívající nedokončené HTTP požadavky),
- -i 10 (interval mezi posíláním dat),
- -r 200 (definuje spojení za vteřinu),
- -t GET (typ odeslané zprávy),
- -l 60 (definuje délku trvání testu),
- -g -o /home/kali/vystup_testu (definuje název a cestu k finálnímu souboru, ve kterém lze zobrazit generovaný graf),
- -u http://<IP_adresa> (definice cílové adresy).

Po spuštění výše uvedeného příkazu začne program SlowHTTPtest generovat definovaný síťový provoz na HTTP adresu WWW serveru. Během chodu programu jsou zobrazovány informace o počtu aktuálního spojení na daný server, statistiky v počtu uzavřených a navázaných spojení a především také status, jestli je daná služba stále dostupná nebo došlo k jejímu odstavení. Sledujte postup aplikace a v případě, že se změní status SERVICE AVAILABLE na NO (nedostupný), ověřte pomocí internetového prohlížeče, že webové stránky jsou opravdu nedostupné.

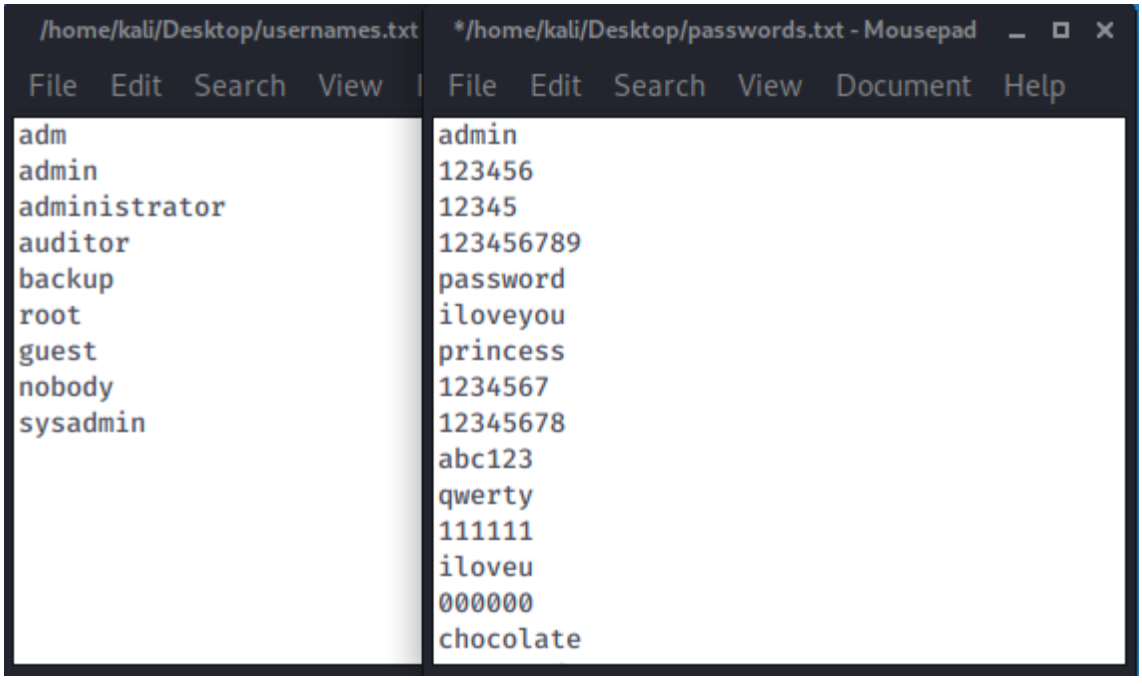
Parametrem -g -o /home/kali/vystup_testu byla nastavena cesta, kde se po ukončení analýzy vytvoří soubor /home/kali/vystup_testu.html obsahující výsledky

analýzy. Pomocí souborového prohlížeče nalezněte tento soubor a pomocí programu Chromium Web Browser jej otevřete. Výsledky testu si zaznamenejte. Odpovězte na otázky 6 a 7 v sekci C.3.6 Doplňující otázky.

C.3.6 Prolomení přístupu na FTP server

Nyní budete mít za úkol pokusit se získat přihlašovací údaje na nalezeném FTP serveru. Pokud vytvořený FTP server nemá nakonfigurované žádné bezpečnostní mechanismy, týkající se například zablokování přihlašovací údajů po několika chybných pokusech o přihlášení, je možné na tento server použít metodu slovníkového útoku pro odhalení přihlašovacích údajů.

K prolomení přihlašovacích údajů slovníkovým útokem je možné využít program **Hydra**, jež je mezi předinstalovanými programy virtualizace KaliLinux. K tomuto účelu byly vytvořeny dva textové soubory umístěné na ploše operačního systému. Soubor *usernames.txt* obsahuje výpis 9 základních uživatelských účtů, které jsou velmi často používány, zatímco soubor *passwords.txt* obsahuje výpis používaných hesel. Zmíněné soubory si můžete prohlédnout například otevřením v aplikaci Mousepad. Ukázka výše zmíněných souborů je k vidění na obrázku C.3.



Obr. C.3: Ukázka souborů obsahující přihlašovací údaje.

Získávání přihlašovacích údajů programem Hydra na adrese FTP serveru je možné zadáním příkazu:

```
sudo hydra -L /home/kali/Desktop/usernames.txt
-P /home/kali/Desktop/passwords.txt ftp://<IP_adresa> -V
```

Po spuštění příkazu se začne program Hydra automaticky přihlašovat jednotlivými uživatelskými jmény a hesly, které bere vždy sestupně z připravených souborů.

Získané přihlašovacích údaje ověřte přihlášením se na odhalený FTP server pomocí internetového prohlížeče zadáním adresy ftp://<IP_adresa>. Výsledky testu si zaznamenejte. Odpovězte na otázky 8 a 9 v sekci C.3.6 Doplňující otázky.

Doplňující otázky

1. Jaký protokol používá aplikace Netdiscover pro detekci aktivních síťových adres?
2. Jaký TCP příznak nesou zprávy, které program Nmap používá při skenování otevřených TCP portů?
3. Jaké příznaky protokolu TCP se při skenování metodou SYN nachází v odpovědi příslušného portu, kterého aplikace Nmap označila jako dostupný (open)?
4. Jaký protokol používá program Nmap při detekci nedostupných (closed) UDP portů?
5. Jaký typ (Type) a kód (Code) zprávy obsahuje protokol z bodu 4 ve své hlavičce, který nese informaci o nedostupnosti UDP portu?
6. V jakém čase konání testu aplikací SlowHTTPtest se stala http stránka nedostupná?
7. Kolik spojení bylo povoleno službou HTTP navázat?
8. Jaké jsou získané přihlašovací údaje na odhaleném FTP serveru?
9. Jaké heslo je uvedené u položky SRV07, které naleznete v souboru report.pdf na odhaleném FTP serveru?

D Obsah přiloženého DVD

- `/` ..... kořenový adresář přiloženého DVD
  - `ciganek_diplomova_prace.pdf` ..... diplomová práce v PDF
  - `LAB1_TCPxUDP` ..... adresář se soubory z lab. úlohy č.1
    - `vzorove_reseni_LAB1.pdf` ..... vzorové řešení lab. úlohy č.1 v PDF
    - `lab1_tcpxudp.gns3project` ..... exportovaný projekt GNS3
    - `po_upraveni.pcapng` ..... soubor se zachycenou komunikací
    - `pred_upravenim.pcapng` ..... soubor se zachycenou komunikací
  - `LAB2_multicast` ..... adresář se soubory z lab. úlohy č.2
    - `vzorove_reseni_LAB2.pdf` ..... vzorové řešení lab. úlohy č.2 v PDF
    - `lab2_multicast.gns3project` ..... exportovaný projekt GNS3
    - `lab2_configs.txt` ..... kompletní konfigurace směrovačů R1-R5
    - `dense_mode.pcapng` ..... soubor se zachycenou komunikací
    - `sparse_mode.pcapng` ..... soubor se zachycenou komunikací
    - `unicast.pcapng` ..... soubor se zachycenou komunikací
  - `LAB3_pentest` ..... adresář se soubory z lab. úlohy č.3
    - `vzorove_reseni_LAB3.pdf` ..... vzorové řešení lab. úlohy č.3 v PDF
    - `lab3_pentest.gns3project` ..... exportovaný projekt GNS3
    - `netdiscover.pcapng` ..... soubor se zachycenou komunikací
    - `TCP_scan.pcapng` ..... soubor se zachycenou komunikací
    - `UDP_scan.pcapng` ..... soubor se zachycenou komunikací
    - `DOS_attack.pcapng` ..... soubor se zachycenou komunikací
    - `FTP_password_attack.pcapng` ..... soubor se zachycenou komunikací